AD-A047 864   MASSACHUSETTS COMPUTER ASSOCIATES INC WAKEFIELD   F/G 9/4
RESEARCH ON INFORMATION SYSTEM SPECIFICATION.(U)
AUG 77 A W HOLT   N00014-76-C-0781

11 B.S.

MASSACHUSETTS
COMPUTER ASSOCIATES, INC.

26 PRINCESS STREET, WAKEFIELD, MASS. 01880 • 617/245-9540

Final Report for Contract N00014-76-C-0781 titled "Research on Information System Specification"

by

Anatol W. Holt

August 9, 1977
CADD-7708-0911

This work was monitored by the Office of Naval Research under Control Number N00014-76-C-0781.

A SUBSIDIARY OF APPLIED DATA RESEARCH, INC.

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)
Massachusetts Computer Associates, Inc. ✓
26 Princess Street
Wakefield, Massachusetts 01880

2a. REPORT SECURITY CLASSIFICATION
Unclassified

2b. GROUP
N/A

3. REPORT TITLE
6 RESEARCH ON INFORMATION SYSTEM SPECIFICATION.

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)
9 FINAL REP[redacted]T. 1 Mar 76-31 May 77,

5. AUTHOR(S) (First name, middle initial, last name)
10 Anatol W./Holt

6. REPORT DATE
11 9 Aug[redacted] [redacted]77

7a. TOTAL NO. OF PAGES
181

7b. NO. OF REFS
0

8a. CONTRACT OR GRANT NO.
15 N00014-76-C-0781 New

b. PROJECT NO.

c.

d. 12 183p.

9a. ORIGINATOR'S REPORT NUMBER(S)
14 CADD-7708-0911

9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
N/A

10. DISTRIBUTION STATEMENT
Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES

12. SPONSORING MILITARY ACTIVITY
Office of Naval Research
Department of the Navy
800 N. Quincy Street, Arlington, Va. 22217

13. ABSTRACT

This document reports the results of the development of methods for system specification and analysis in conjunction with the National Software Works Project currently in progress in our company. This report indicates that the ideas developed and partially tested in this setting are, potentially, of wide utility in the description and analysis of systems where the interest focuses on the relation of communication among a set of agents with interdependent tasks.

There is a description of a theoretical framework in which to study the effects of the propagation of information in a system context as a result of analysis of logical dependence relations in nets. There is a discussion of the relationship between the concepts concurrency and choice, a survey of the use of Petri-nets for the representation of organizational relations.

Unclassified
Security Classification

LB

093 745

Unclassified

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Information						
Nets						
Petri-nets						
Communication						
Concurrency						

Unclassified

Security Classification

TABLE OF CONTENTS

Approach to Practice

Theory

Related Papers

Appendices

I Introduction

I Introduction

The research of the Information System Theory Project under ONR contract NOOO14-76-C-0781 in the period March 1, 1976 to May 31, 1977 was divided into two major parts as described under C1 and C2 of the proposal - included here as appendix XA.

The work on C1 - "developing new tools of demonstrated utility for determining, expressing and manipulating information system specifications" - was not carried out in conjunction with a Navy application, as suggested under D1 of the proposal because of difficulties in achieving a mutually acceptable definition of the task between ourselves and The Office of Naval Research, Information Systems Branch of the Department of the Navy in Arlington, Virginia. We had originally intended to carry out a study of the 3M system, as per appendix XB. In lieu of this study we developed some methods for system specification and analysis in conjunction with the National Software Works Project currently in progress at our company. We believe that the ideas developed and partially tested in this setting are, potentially, of wide utility in the description and analysis of systems where the interest focuses on the relation of communication among a set of agents with interdependent tasks. Our results on this front are reported as Chapter II of this report.

The work on C2 - "Analysis of logical dependence relations in nets" - resulted in the development of a theoretical framework in which to study the effects of the propagation of information in a system context. The general view of "information" underlying this framework is the following.

A1 The local effect of the arrival information at some place and time within a system is to determine the outcome a well-defined choice at that place and time. In general, information from a multiplicity of sources must "fan in" to a given place at a given time to determine the outcome fully. In so coming together, these separate contributions to choice resolution are synchronized.

The outcome of a choice may be viewed as new information, separate protions of which "fan out" in order to make their contributions to determining the outcome of subsequent choice.

> Thus the propagation of information in a system is realized in the form of <u>connections between the resolutions of local choices.</u>

The theoretical framework is described in this report under Chapter III.

Finally, Chapters IV and V are two papers which were prepared under this contract. The first is a working paper which discusses the relationship between the concepts <u>concurrency</u> and <u>choice</u>. The issues which it discusses are extracted from the work described in Chapter III. The second contains a survey of the use of Petri-nets for the representation of organizational relations - a survey produced as an invited paper for the Gesellschaft fuer Informatik annual meeting, Stuttgart, 1977.

II <u>The Representation of Intercommunication in a System Context</u>

II The Representation and Analysis of Intercommunication in a System Context.

A Introduction

In this section we introduce a method for describing and analyzing the relationships between a set of intercommunicating entities, in a systems context. These intercommunicating entities might be implemented as programs, hardware processors and/or human agents engaging in communicative activity.

We shall describe our development as the rudiments of a design and specification language DSL. We shall demonstrate a real design problem in the area of inter-process communication which can be helped by simple analytic techniques applicable to expressions in DSL. Such techniques could be carried out by computer-implemented algorithms operating on DSL expressions as data.

Our descriptions do not constitute a finished proposal for DSL and a related package of algorithms. Such a proposal would require much more detailed , formal descriptions than those offered below. On the other hand, more detail and more formality is not warranted until the ideas presented here have been exercised sufficiently in the context of practical applications.

The example material used in the development of DSL came from a major software project - The National Software Works, NSW for short - currently in progress at Massachusetts Computer Associates, under ARPA and US Air Force sponsorship. The purpose of this project is the construction of a software production environment which enables software builders to use facilities that are distributed over the ARPANET. The NSW design effort naturally brought all the design and implementation difficulties to light which are characteristic of distributed control processing - problems of relative timing, resolution of conflicts over access to resources, problems of identification (of messages, files, processes, etc.), recovery in the face of communication, hardware and software failures, etc., etc. In facing these problems COMPASS became acutely aware of the need for something like DSL and associated algorithms.

A precursor of DSL - called "Scenario Language" - (also an outgrowth of our research) was used in certain phases of NSW design. Scenario Language is briefly discussed in section E of this chapter.

From the theoretical point of view, DSL is based on prior work on Petri-nets, and especially the role/activity interpretation developed in ca 1975. (This interpretation is discussed in section E of Chapter V.)

B General Characteristics of DSL

DSL has been principally conceived as a graphic language, and it is in that form that it is demonstrated and discussed in sections C and D below. We would also expect to develop derivative forms consisting of ordered sequences of statements and declarations - like the forms that are typical of programming languages.

DSL is to be applicable to all levels of system descriptions - from the grossest to the finest. At every level, however, it remains concerned with communication - i.e. message flow from entity to entity in system space.

DSL, in its basic definitions, does not impose any particular design principles - i.e. general restrictions in the forms of interprocess communication, or general restrictions in the forms of building blocks out of which processes are constructed. It should, however, lend itself to the specializations which would follow from such impositions. Adherence to such restrictions will naturally result in useful notational shorthands, such as those used in scenario language. It is expected that DSL will permit the development of such restrictions and related notational conveniences. Thus DSL might be viewed as a basis for the development of special design disciplines.[1]

DSL integrates smoothly with yet more basic forms of expression which have been studied in Project ISTP. In connection with these forms there exist certain mathematical analytic techniques (e.g., the theory of marked graphs [6]) which, in the longer run, may be turned to good account in the form of DSL algorithms.

DSL lends itself to the description of the expected behavior of system users just as it does to the description of computer supported processes.

A means of system description such as DSL, one well enough formalized, can become input to a DSL processor, DSLP, implemented, for example, on a package of program for a computer. Such a processor would have the following major functions.

[1] Under the general label "structured design" there have recently been developed a number of techniques - in some cases with supporting software - by various groups in the USA (see [7, 8, 9] for examples). None of these techniques, however, focus on communication, and none of them have any significant theoretical support. All of them are instead extensions of the ideas that were developed to describe program architecture. Thus the phrase 'structured analysis' is itself a natural follow-on to the phrase 'structured programming'.

We now turn our attention to DSL.

DSL is expected to offer the following general capabilities:

B1 .1 to accept expressions in DSL as input from a terminal - all keyboard, or keyboard and graphic combined.

.2 to store these expressions in a design and specification data base

.3 to extract from, assemble, condense the stored expressions to form new expressions for display and/or storage. These capabilities would have a variety of uses, such as:

- Obtaining representations of a system which are variously focused - e.g. focused on the histories of messages, or on the histories of communicating processes.

- Obtaining representation of the same material at various levels of detail.

- Obtaining representations which restrict ones attention to a selected sub-portion of a particular design.

.4 giving computational aid in checking for well-formedness of description, consistency of behavior of independent but communicating processes, deadlock free-ness, liveness, adequacy of buffer capacities, transmission and processing rates, adequacy of source and destination identification in message addresses, correctness of assumptions (such as: only one message from a certain source can be in a certain buffer at a certain time) etc., etc.

- The power of such computational aids will depend in part on the structural design principles to which the user is willing to adhere, and in part on the size of the system portion which is isolated for checking. For example, while no general algorithms will be feasible for verifying that a total system is free of deadlocks or critical races, such checks will be practically feasible in restricted contexts (see section D below).

C The Nationa: Software Works and the Design Language

This section has two purposes: to give an over-all description of NSW (National Software Works) and to demonstrate the character of DSLA. Having accomplished both of these purposes, we shall then be in a position to give examples of the problem types which might be aided by a DSLP.

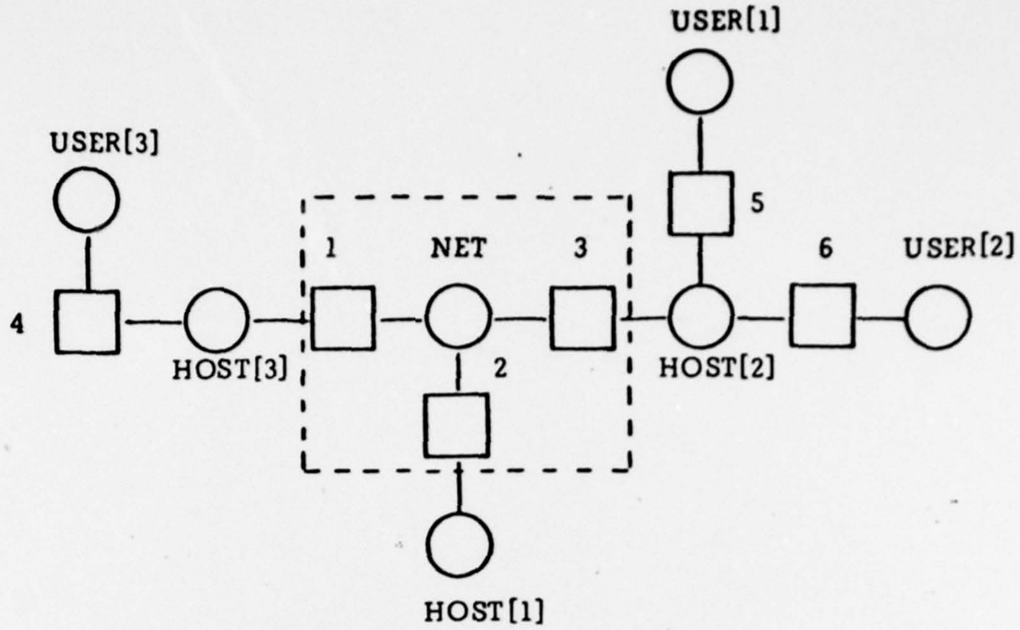
In 1973, COMPASS began work on a system known as NSW, under ARPA sponsorship. The primary purpose of the system is to provide a window through which programmers can access all software, hardware, and data resources available on the ARPA NET in a uniform manner. Via NSW, the programmer can specify computer operations to be performed requiring file and programming resources which are initially scattered over a number of host computers on the NET. The execution of such an operation would therefore require the movement of files from one host to another, with automatic format and name transformations, as required by the host environments.

To a programmer operating via NSW, the entire ARPANET appears, in effect, as a single giant computer with NSW as its operating system.

In addition to controlling a great diversity of computing resources at many locations, NSW is also designed to handle a large number of concurrent users operating on many different host computers. All of these NSW users may access common files, and NSW must control access rights and resolve priority conflicts.

Another important aspect of the NSW design problem is system reliability and survivability in the face of hardware and software crashes. In one design plan, the catalogue of NSW files - called the NSW data base - is maintained concurrently on a multiplicity of hosts. These data base copies must be updated in a mutually consistent manner. If, at one site, the data base cannot be maintained current because of some temporary failure, it should be possible to "catch-up" and bring the data base back in line when the failure is repaired. With this general introduction we now pass to exhibiting the NSW problem with the help of DSL.

C1



C1 pictures, by way of example, the basic operational framework for NSW. It does this with the help of two types of symbolic (and conceptual) elements:

.1 ○^{r} representing resources

.2 □^{a} representing activities

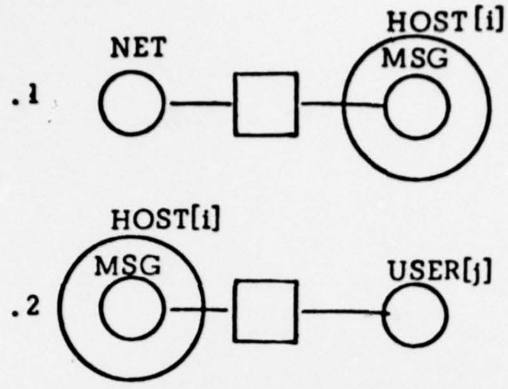
and a relation .3 ○^{r}—□^{a} meaning resource r is necessary to activity a (or equivalently, activity a requires resource r)

The three hosts in our picture represent computer systems with the appropriate hardware and software capabilities to be viewed as NSW hosts. Each of them interacts with a network of communication facilities - called the NET. The activities 1, 2, 3 are essentially message passing activities. With the NET pictured as an explicit resource, messages will be seen as travelling from a host to the NET or from the NET to a host rather than directly from host to host.

Connected to some, but not all, of the NSW hosts are NSW users who communicate with their respective hosts. A host to which no user is directly connected will contain program and file resources which NSW users can access indirectly via the NSW system.

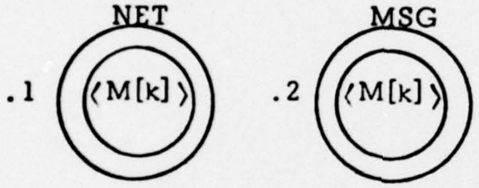
The dashed-line which surrounds the NET and activities 1, 2, 3 represents a 'higher level' communication activity in which hosts are seen as interacting with one another. The NET would then be viewed as a resource internal to that activity.

C2



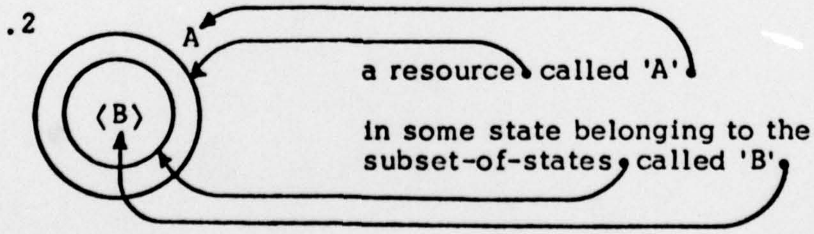
Each NSW host must contain a capability (or *resource*) called MSG through which all NSW message traffic flows. As is shown in .2, this includes message traffic between the host and an NSW user, if any exist.

C3



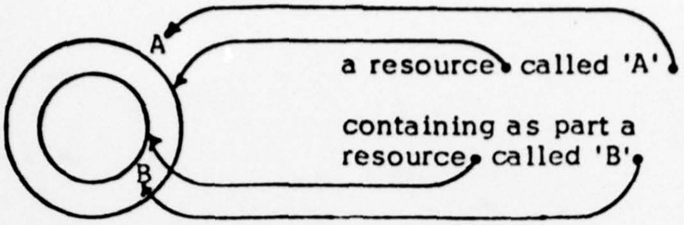
C3.1 is a picture of the NET *holding* a message M[k]; similarly, C3.2 shows MSG *holding* a message. Here follow some remarks about the holding relation.

We think of messages as resources. Certain classes of resource relevant to the operation of a system may be *present* or *absent* at some places and some times. C3.1 and C3.2 assert that messages *may* be present in the net and in MSG; but they *may* also be absent at these places. What it comes down to is this: *MSG in certain of its states is interpreted as holding a message*, *but in others of its states is interpreted as not doing so* (and the same with the NET). We can translate this explanation into the syntactic detail of C3.1 and C3.2.



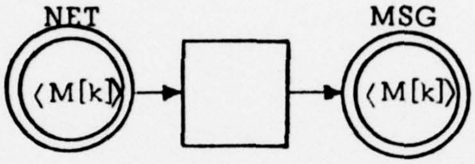
Compare this to:

.3



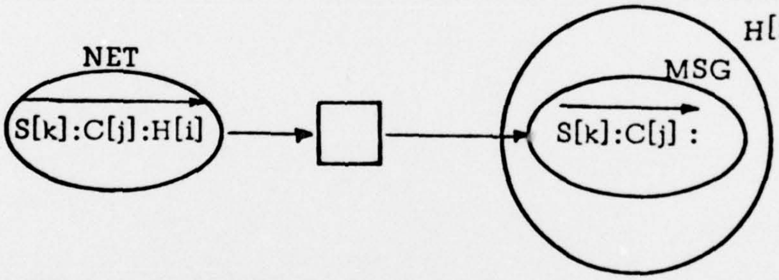
Both .2 and .3 are interpretable as the presence of a resource B "in" (resource) A : but the first picture (.2) also allows for its absence, while the second picture (.3) does not.

C4



This picture represents the transfer of a message from the NET to MSG. The NET and MSG both participate in the activity. The arrow-heads add the following understandings: a message M(k) held in the NET is a resource required as an input to the activity and the message M(k) held in MSG is required as an output of the activity. In addition the following is implied: at output time the message M(k) is no longer held in the net, and at input time it is not yet held in MSG. Thus the arrow-heads imply state-change in both the NET and in MSG.

C5

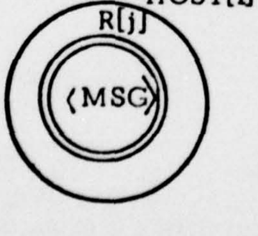


The oval form means the same as a circle.

Here we show a message formatted into three components: S[k], the source address; C[j], the contents; H[i] the destination address - in this example, NSW host i . The arrow which overlines the three components is our standard symbol for a message. Since messages will always be seen as held in some place, we can omit the context '⟨ ⟩'' which was explained above. "MSG' is not part of the destination address because every NSW host connects to the net via MSG. The destination address is not shown in the message held in MSG because it is no longer relevant.

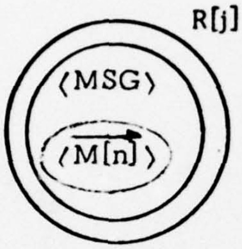
C6

.1



This picture should be viewed as a modification of the right half of C2.1. In all envisaged computer systems the MSG capability will exist in the form of an active program and associated tables. Thus the computer system will have to devote some portion of its resource R[j] to holding MSG. The picture C6 sets the stage for activities which result in the destruction of the MSG capability without the attendant destruction of the NSW host. It also sets the stage (should this be important) for activities resulting in the move of MSG from one place to another within the host. By implication, we have now illustrated two levels of the holding relation: computer resources holding MSG; MSG holding messages, as explicitly shown in .2 below.

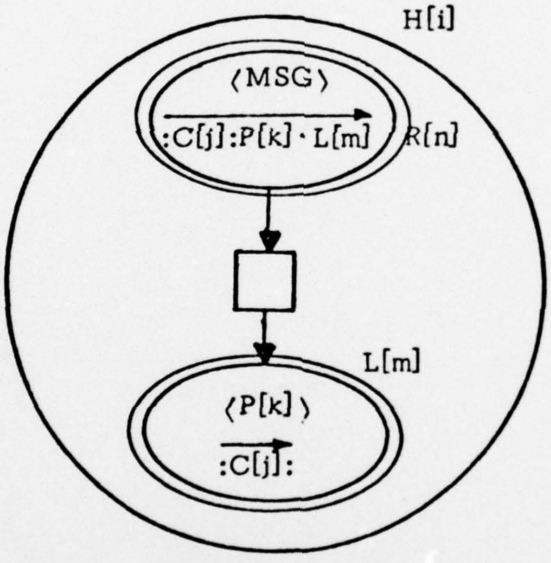
.2



R[j] - a resource of the host suitable for holding MSG

C7

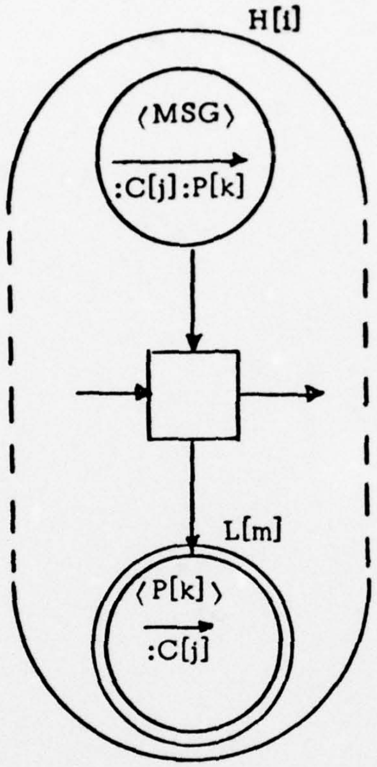
.1



Here we see MSG transmitting a message to an NSW process identified in the address of the message as 'P[k]·L[m]' . 'P[k]' is an NSW generic process name, while L[m] is an addressable location within the host where an instance of the generic process resides. In NSW 'P[k]·L[m]' would be called a specific process identifier (in contradistinction to 'P[k]').

MSG is also capable of acting on messages which contain a generic identifier only as destination address. In some cases it will find an existing instance of the generic process capable of accepting the message; in other cases it will cause the "creation" of a new instance, initialized to process the message. Our next diagram pictures an activity which would be interpretable in either of these mentioned ways.

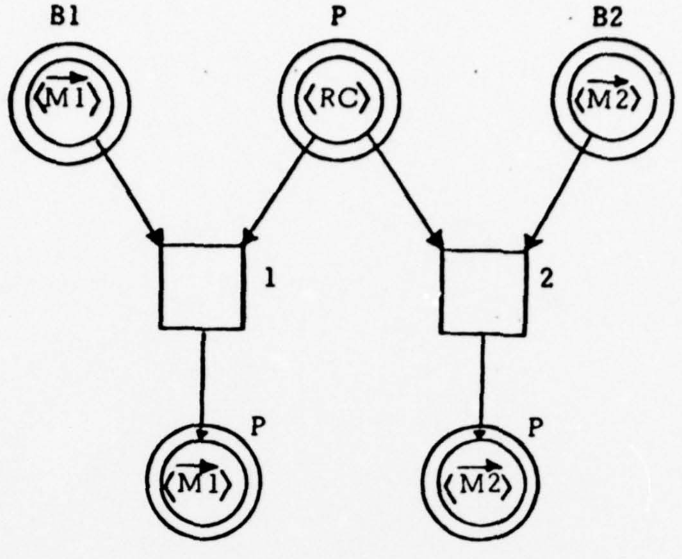
.2



The result of the activity in .1 and in .2 - as far as the result is explicitly shown - is the same. But the activity in .1 is just that of <u>message transmission</u> - related to the formal feature of C7.1 that the output consists exactly of the resource which is named in the destination address holding the content of the message. In C7.2 the output also consists of the resource which is named in the destination address of the message holding the content, but there is an additional resource involved as well - namely L[m] . Information additional to that in the message is required to locate (or construct) a message holder of the form required for message delivery.

C8

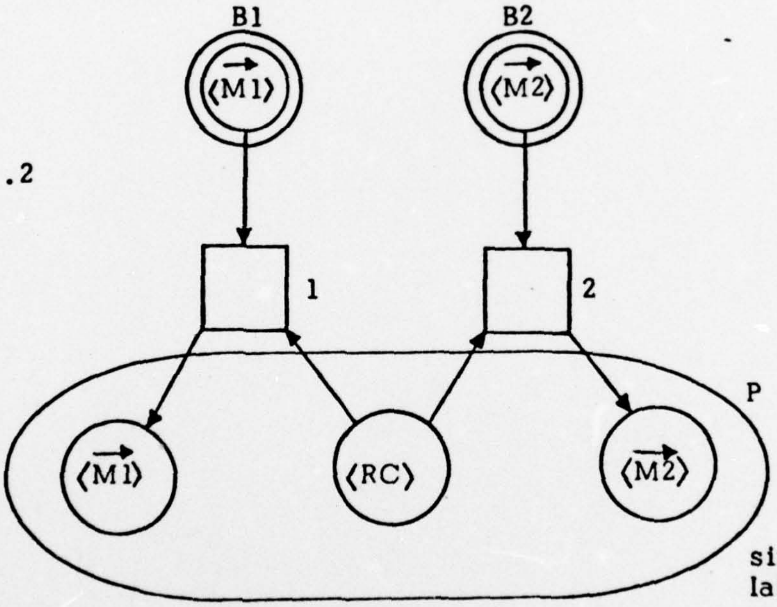
.1



P - process
B1 - buffer 1
B2 - buffer 2
RC - 'ready for communication'

or equivalently:

.2



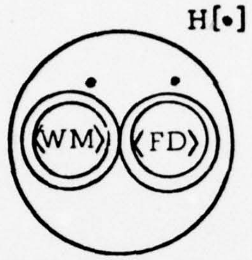
since all three circles labelled 'P' in .1 represent the same resource.

Our pictures represent a process P at a stage RC when it is prepared to receive one of two messages $\overrightarrow{M1}$ and $\overrightarrow{M2}$. As far as the process is concerned it is to undergo one of two state transitions: from holding RC to holding $\overrightarrow{M1}$, or from holding RC to holding $\overrightarrow{M2}$. On the assumption that $\overrightarrow{M1}$ and $\overrightarrow{M2}$ are distinct, these transitions must be mutually exclusive. Thus, a difficulty may arise if messages M1 and M2 are available in their respective buffers at the same time. If no additional resource is available for arbitrating between activities 1 and 2 no well-defined behavior on the part of the process can take place.

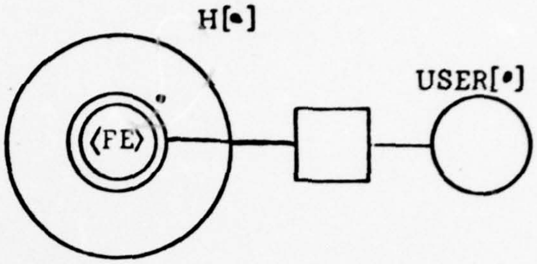
This section now concludes with two more items about NSW which are preparation to discussing some detailed design and associated problems.

C9

.1



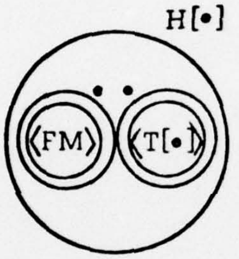
.2



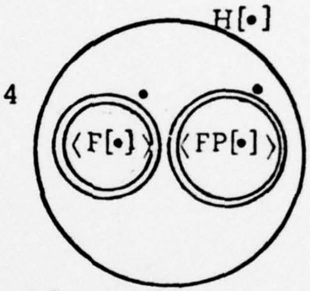
WM - Works Manager
FD - NSW File Directory

FE - Front End

.3



.4



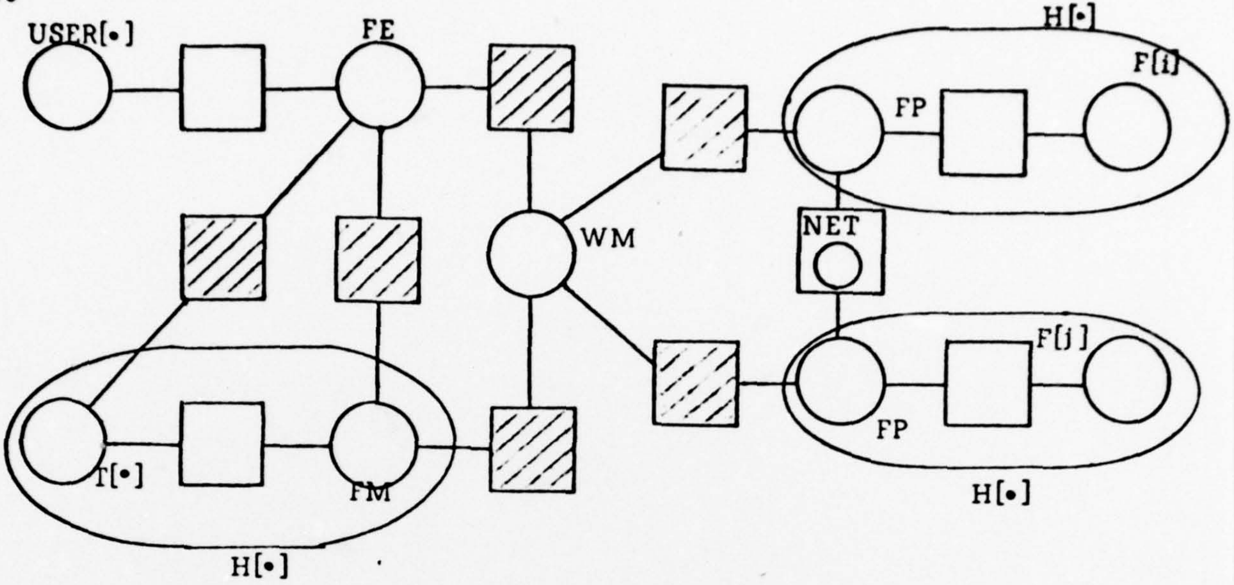
FM - Foreman
T[•] - NSW tools

F[•] - NSW files
FP[.] - File Package

(In these pictures dots replace identifiers whose identity is a matter of indifference to the picture.)

Comments: All NSW processes (other than MSG) are divided into four classes: Works Manager, Front End, Foreman, and File Package processes. Our pictures show various host capabilities which go together. For example a host which has a WM capability (i.e., supports WM processes) must also have an NSW File Directory which is referenced and updated by WM processes. A host with which users communicate directly must support FE processes which mediate between the user and the rest of the NSW world. A host which supports NSW tools must also support FM processes which mediate between tools and the rest of the NSW world. Finally, a host which supports NSW files must also support a File Package processes which mediate between some collection of NSW files and the rest of the NSW world. (As described earlier, MSG mediates all NSW communications and must be present on every NSW host. Hereafter we shall omit all reference to MSG, unless explicitly required.)

C10



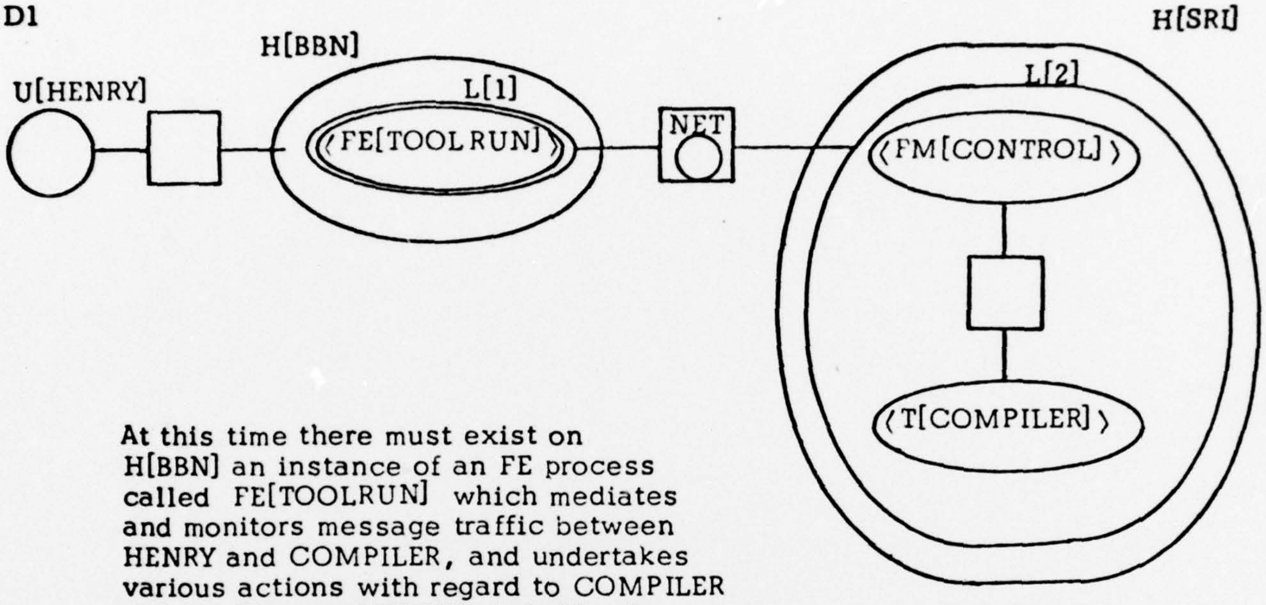
This picture encompasses all possible NSW communication patterns. All of the shaded activity boxes represent either inter-host or intra-*host communications depending on the host identities which contain* the various resources shown. The two File Package capabilities shown are, by implication in different hosts since they communicate over the net. Theoretically it is possible that all NSW files reside on a single host, in which case one of the two file-bearing hosts together with all of its connections would be deleted. Thus, theoretically, the entire NSW system could be confined to a single host.

D Example of Detailed NSW Design and Some Associated Problems for Analysis

We are now ready to describe some specific NSW mechanisms and to illustrate investigations of their properties which might be carried out with the help of a DSLP.

Let us suppose that user HENRY is in the midst of running a tool *COMPILER. HENRY is on host BBN;* COMPILER is on host SRI.

D1



At this time there must exist on H[BBN] an instance of an FE process called FE[TOOLRUN] which mediates and monitors message traffic between HENRY and COMPILER, and undertakes various actions with regard to COMPILER and its files on HENRY's behalf. On H[SRI] there must similarly exist an instance of an FM process called FM[CONTROL] which controls the use of COMPILER.

L[i] are locations that can be addressed in messages.

FM[CONTROL] and FE[TOOLRUN] each contain a buffer where incoming messages are stored for processing. For the sake of simplicity we shall assume a single buffer for intra- as well as inter-host messages. We shall also assume that these buffers have sufficient capacity to accomodate all incoming messages, given the expected rates of message arrival and message processing.

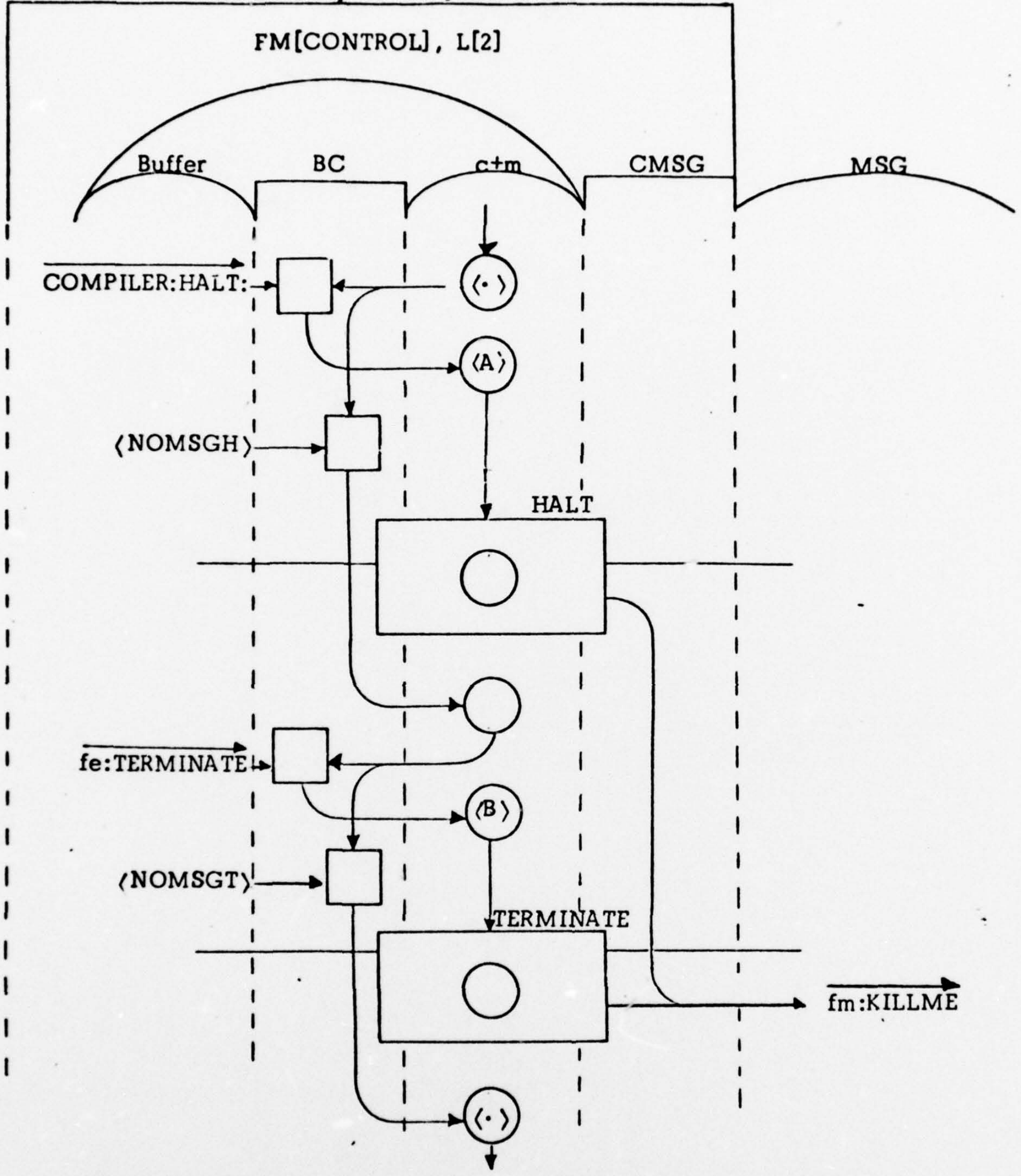
Quite generally NSW processes are message driven. More exactly, they are message and timer driven - since, when expected messages fail to arrive, timer signals will drive the processes onwards.

The diagrams D2 and D4 - D8 which follow exhibit, in some detail, the portion of FE[TOOLRUN] and also the portion of FM[CONTROL] which pertain to ending the operation of the tool, COMPILER. From this material we shall extract a diagram which exhibits the joint behavior of FE[TOOLRUN] and FM[CONTROL] in ending the operation of the tool. This extraction is our first example of what the proposed design tool DT1 will accomplish algorithmically. Following this, we shall subject the diagram of joint behavior to some analysis for communication problems. This analysis is our second example of a function which DT1 would be expected to perform.

To understand the extraction process and its subsequent analysis it is not essential that reader understand the exhibited portions of FE[TOOLRUN] and FM[CONTROL] in detail, although all the material needed for that understanding is in fact provided. It is enough just to "get the idea of how these descriptions go". (These same portions FE[TOOLRUN] and FM[CONTROL] are re-described in a more condensed form in section E.)

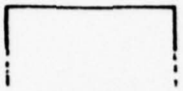
A tool run may end because the tool has reached the normal end of its operation, or because the user (or FE[TOOLRUN] on the user behalf) requires that the run be ended. After FM[CONTROL] has taken the necessary ending action, it notifies MSG that it itself has ended, and goes into inactive status. (The resource L[2] is then available for re-use.)

D2 the activity of FM[CONTROL]

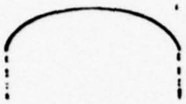


COMPILER:HALT: a message from COMPILER declaring it has halted
fe: FE[TOOLRUN]·L[1]·H[BBN]
fe:TERMINATE: a message from FE[TOOLRUN] that the COMPILER run is to be terminated
NOMSGH: absence of a HALT message
NOMSGT: absence of a TERMINATE message
c+m: process control + memory

In D2 and further diagrams below, the heading information groups the material in the diagram into resources and activities. Everything embraced by



is an activity. Everything embraced by



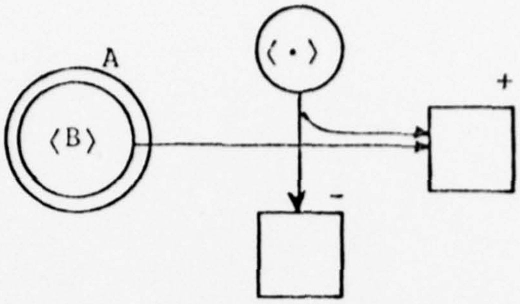
is a resource. At the grossest level, therefore, our diagram shows the FM[CONTROL] activity which requires the MSG resource - a resource which is external to that activity. Internal to the activity is the resource FM[CONTROL] in message-addressable location L[2] . That resource is shown as subdivided into two resources: a buffer and process control + memory, c+m. These two are shown as connected to one another by an activity BC which is internal to the FM[CONTROL] resource, while c+m is also connected to MSG by an activity CMSG. The HALT and TERMINATE activities each belong partially to BC and partially to CMSG, since they require both the buffer and MSG. Furthermore each of these activities has an internal resource which is shown as belonging to c+m .

Each of the resources internal to c+m represent subsets of states of c+m which must be mutually exclusive to one another (since the activities they enable are mutually exclusive to one another). Thus one can tell at a glance that D2 represents a sequential process (except for possible concurrencies internal to HALT and TERMINATE). The activities which transfer messages into the buffer are not shown. The various contents of the buffer are not encircled in any particular way so as to leave open the question as to how many message locations internal to the buffer are involved. (Knowledge of the structure of the activities in which a buffer in involved will, under various auxiliary assumptions, allow one to compute the minimum and/or maximum buffer size which will support the activities in question.) The specific labels A and B have been inserted in two circles internal to c+m to allow reference to them in subsequent diagrams.

D2 represents only a small portion of FM[CONTROL] , as indicated by the stubby arrows at the beginning and end of c+m . It represents only the stretch of activity where the possibility of tool ending is being examined.

In D2 we have two examples of tests for the presence or absence of a message in the buffer. (See C3 for first mention of presence and absence.) Such presence/absence tests are a constantly recurring phenomenon in communication systems, and always imply reference to a clock. What is being tested is: has message m arrived since the last arrival opportunity, and prior to the expiration of some deadline? In the case of the two such tests in D2, the deadline expiration occurs when the time has come for the execution of the test instruction, as determined by the computer clock. In D4 - detailing the HALT activity shown in D2 - there are presence/absence tests in which the deadline expiration is governed by a timer. For presence/absence tests we shall adopt the following conventions:

D3

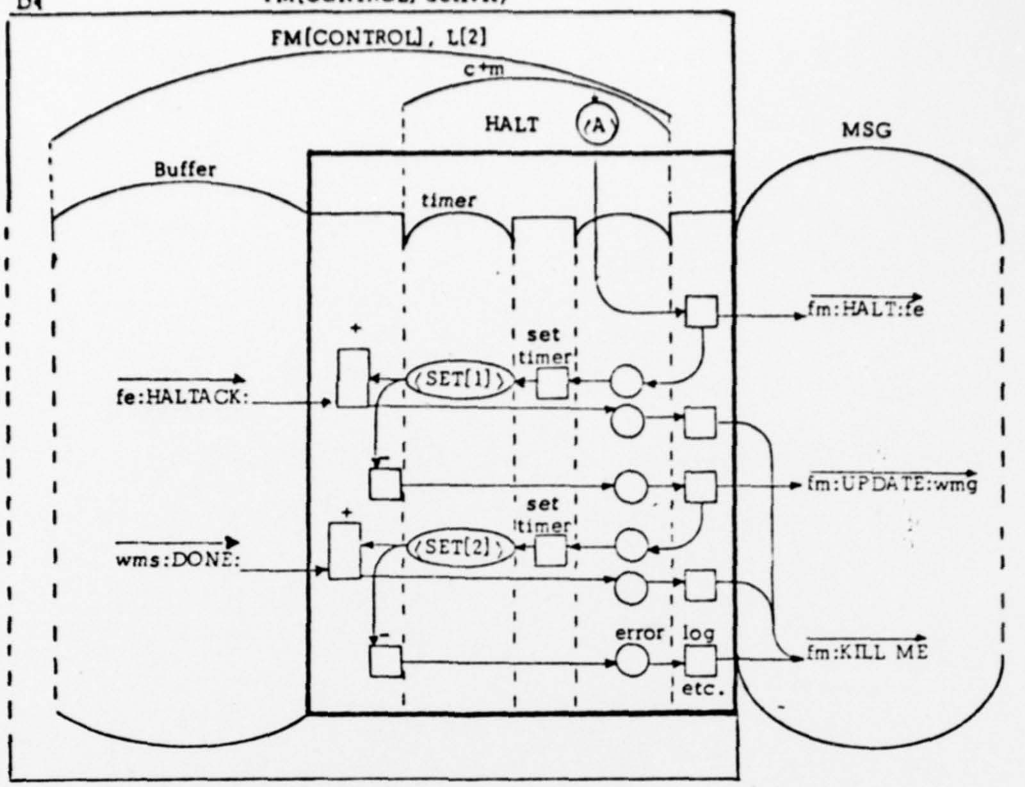


Test for the presence of B in A

conventions: (1) the '+' and '-' labels on the activity boxes
(2) the omitted input from A to the activity labelled '-'.

We shall now exhibit the HALT activity in D2 in more detail.

D4 FM[CONTROL] activity

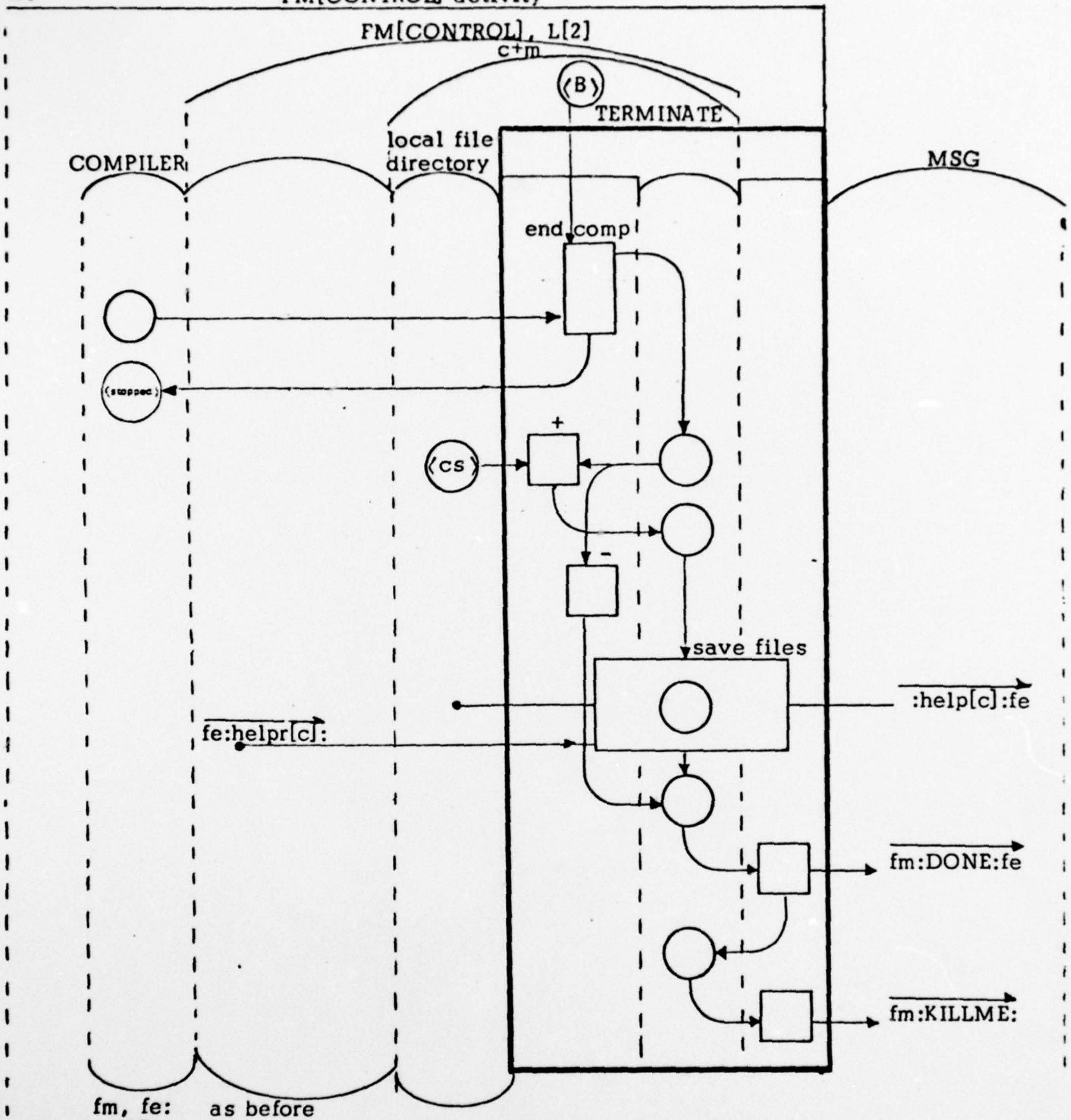


fe: FE[TOOL RUN]·L[2]·H[BBN]
fm: FM[CONTROL]·L[2]·H[SRI]
wmg: WM[a]
wms: WM[a]·L[b]·H[c]
} 'g' for <u>generic</u>; 's' for <u>specific</u>

The HALT process begins by sending a message to HENRY's FE that the COMPILER has halted. A timer is then set to wait for an acknowledgement. If the acknowledgement comes in time, MSG receives notice that FM[CONTROL]·L[2] has ended.[1] If not, it is assumed that FE[TOOLRUN] ·L[1] did not get the message because of any of many possible failures, and a message is sent to a WM process that can update the WM database to reflect the end of the COMPILER run. (Under normal circumstances FE[TOOLRUN] notifies the WM that the COMPILER run has ended.) Once again a timer is set to await a WM acknowledgement. If it fails to come, an error is logged.

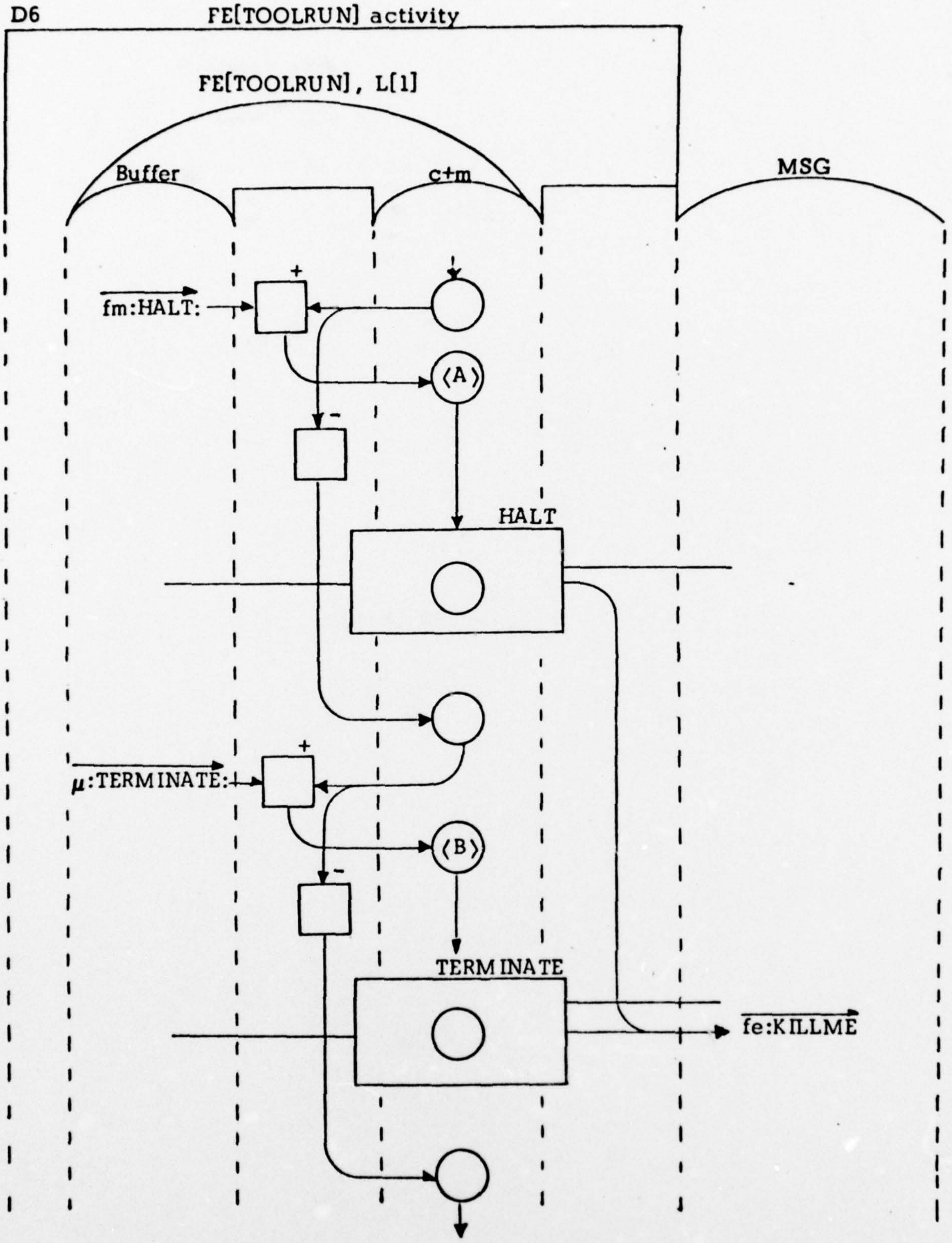
[1] Comment on timer operation. As output of the set timer activity we have a timer that has been set to some expiration time and is now running. In both instances shown, the timer is shown as interacting with the buffer. If the message awaited is found in the buffer after the timer is set and before it expires then the '+' activity takes place. It's output is the return of control to the waiting process. Other parts of the output condition that are implied are: the timer turned off and the message removed from the buffer. The '-' activity takes place if the timer runs to its expiration time in the <u>absence</u> of the expected message in the buffer. Depending on the implementation, arbitration may be required as between the '+' and '-' activity if the message enters the buffer "just as" the timer reaches expiration (see C8).

D5 FM[CONTROL] activity



fm, fe: as before
⟨CS⟩: files produced by COMPILER that may be worth saving
endcomp: end compiler operation
help[c]: FM[CONTROL] asks for help in deciding what files to save
helpr[c]: the help[c] response from fe

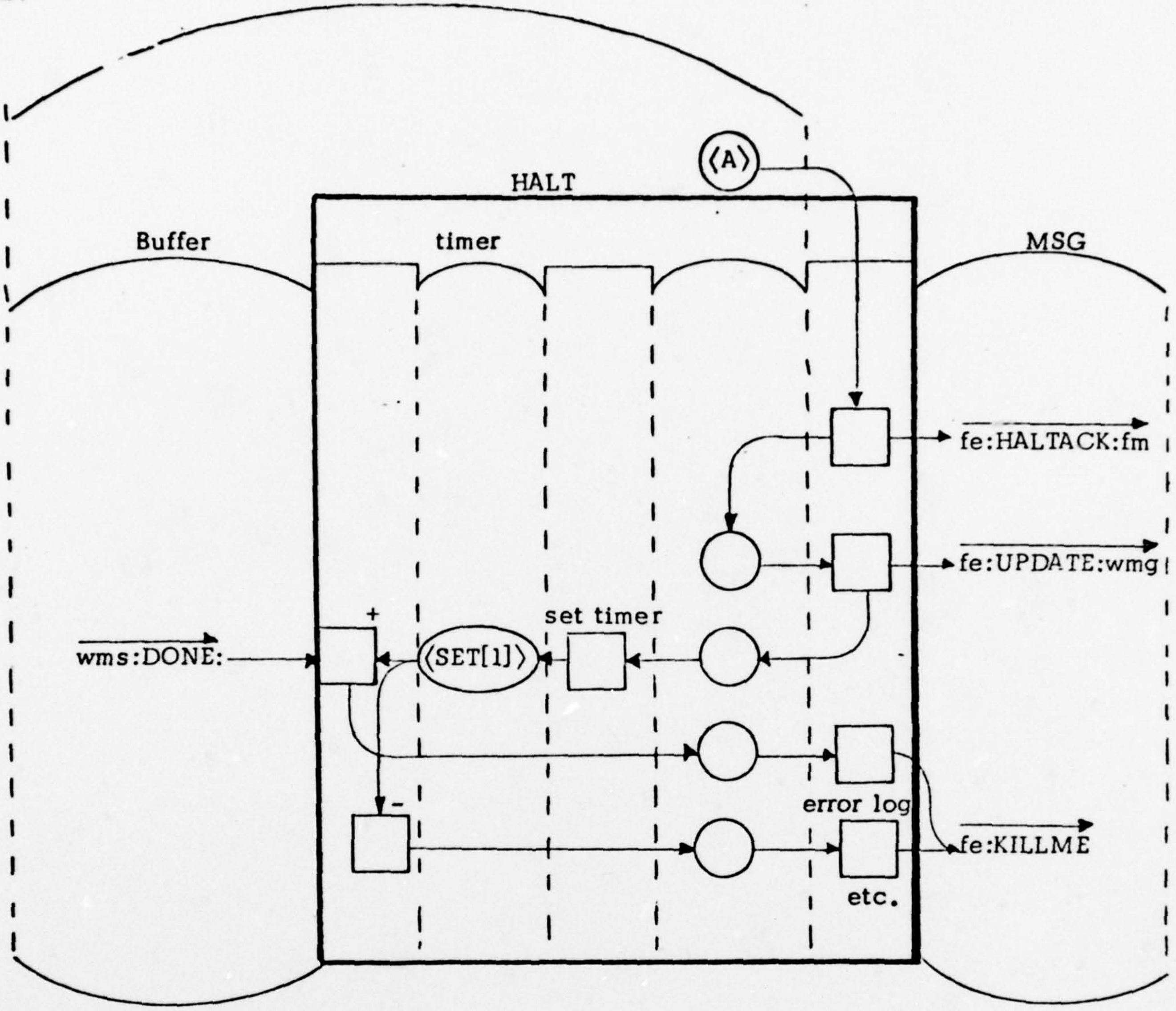
D2, D4, and D5 have their FE counterparts which follow below.



fm, fe: as before
μ: HENRY

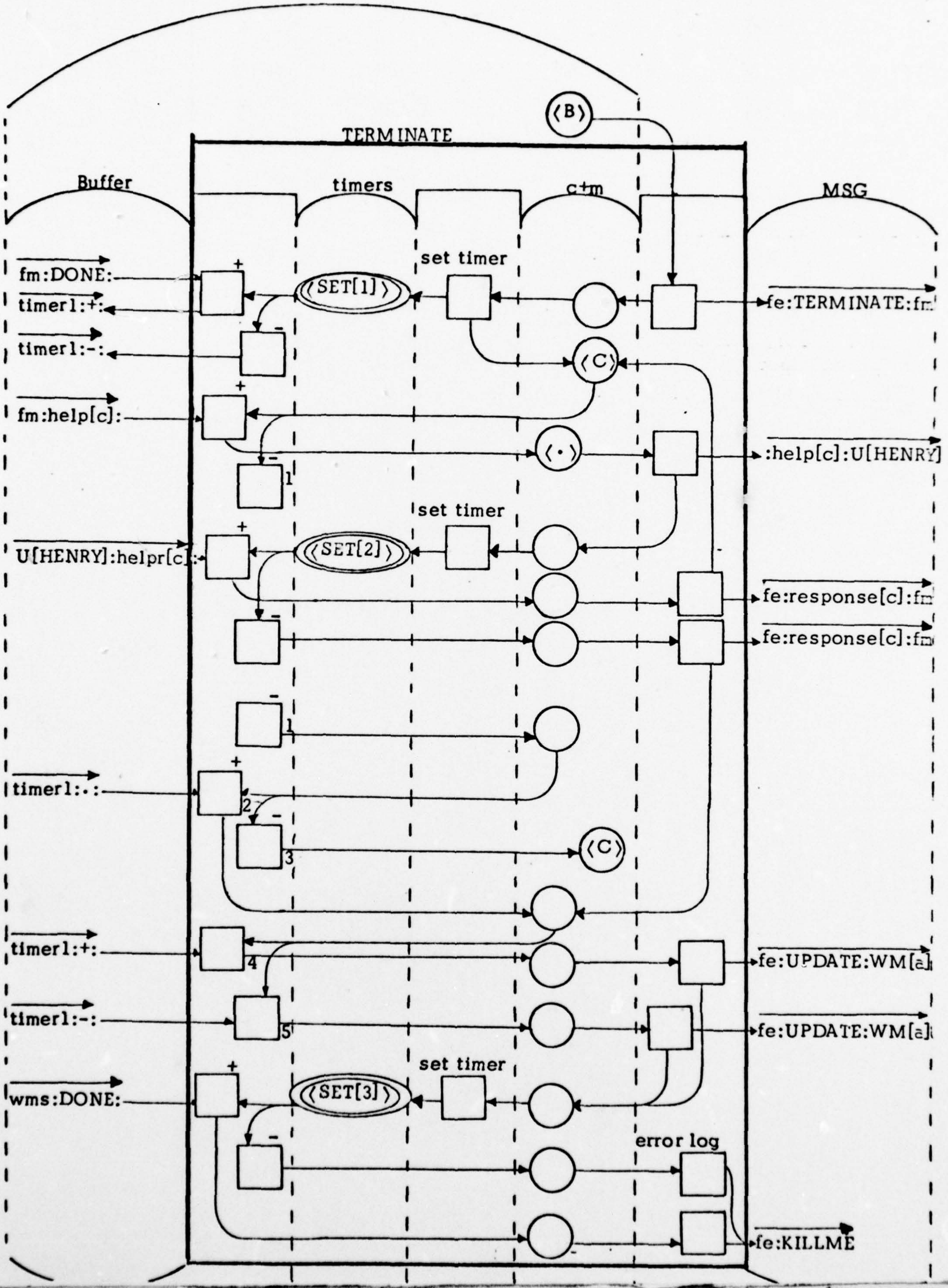
D7

FE[TOOLRUN], L[1]



fe, fm: as before
wmg, wms: as in D4

D8 FE[TOOLRUN], L[1]



Comments: The first timer set in this diagram (SET[1]) functions differently from the timers in D4. This time the process is not "put to sleep" while the timer runs, but continues - starting with state C - while the timer runs. The result of test for presence or absence of fm:DONE is stored in the buffer in the form of message from the timer to the process. The activities marked '2' and '3' represent a test for the presence or absence in the buffer of a message from the timer. The activities marked '4' and '5' represent a test for whether, by expiration time, the 'fm:DONE' message was in the buffer or not.

In diagram D8 we have made explicit use of the following general convention: identically labelled resources or activities in identical contexts are to be understood as identical. Thus, within c+m there are two circles, both labelled 'C'. They are to be understood as two representations of c+m in the subset of states called 'C'. Equivalently, we could have drawn an arrow from the upper one of these two circles to the activity box labelled '3'. The same convention applies to the two occurrences of activity boxes labelled '1', the two occurrences of 'timer1:+:' etc.

We now want to extract the joint behavior of FE[TOOLRUN] and FM[CONTROL] with respect to ending the tool run, as implied by the six descriptions D2, D4 - D8. We shall develop a diagram of this joint behavior under the following special assumptions.

D9

.1 The joint behavior of the two processes may be constrained by message flow mediated by other processes (such as the Works Manager). Such constraints, if they exist, will be explicitly left out of account.

.2 We wish to focus our interest on the relations between FE[TOOLRUN] and FM[CONTROL] on the assumption that all expected messages arrive in time. Therefore, all behaviors which follow upon expirations of timers will be omitted.

.3 We shall take account of the internal activity of each process only insofar as it may affect the other. As far as FE[TOOLRUN] is concerned, the user HENRY will be regarded as internal to it.

Comment: if the extraction process about to be shown were carried out by DT1, the assumptions D9 would be expressed formally as parameters to DT1.

The extraction depends upon the application of the following rules.

D10

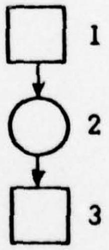
.1 All inputs to FE[TOOLRUN] other than messages from FM[CONTROL] are deleted.

.2 All inputs to FM[CONTROL] other than messages from FE[TOOLRUN] are deleted.

.3 All paths passing through boxes marked '-' are deleted.

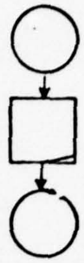
After these deletions

.4 All instances of



where 1 produces 2 only
2 is produced by 1 only
2 is input to 3 only
3 has no inputs besides 2

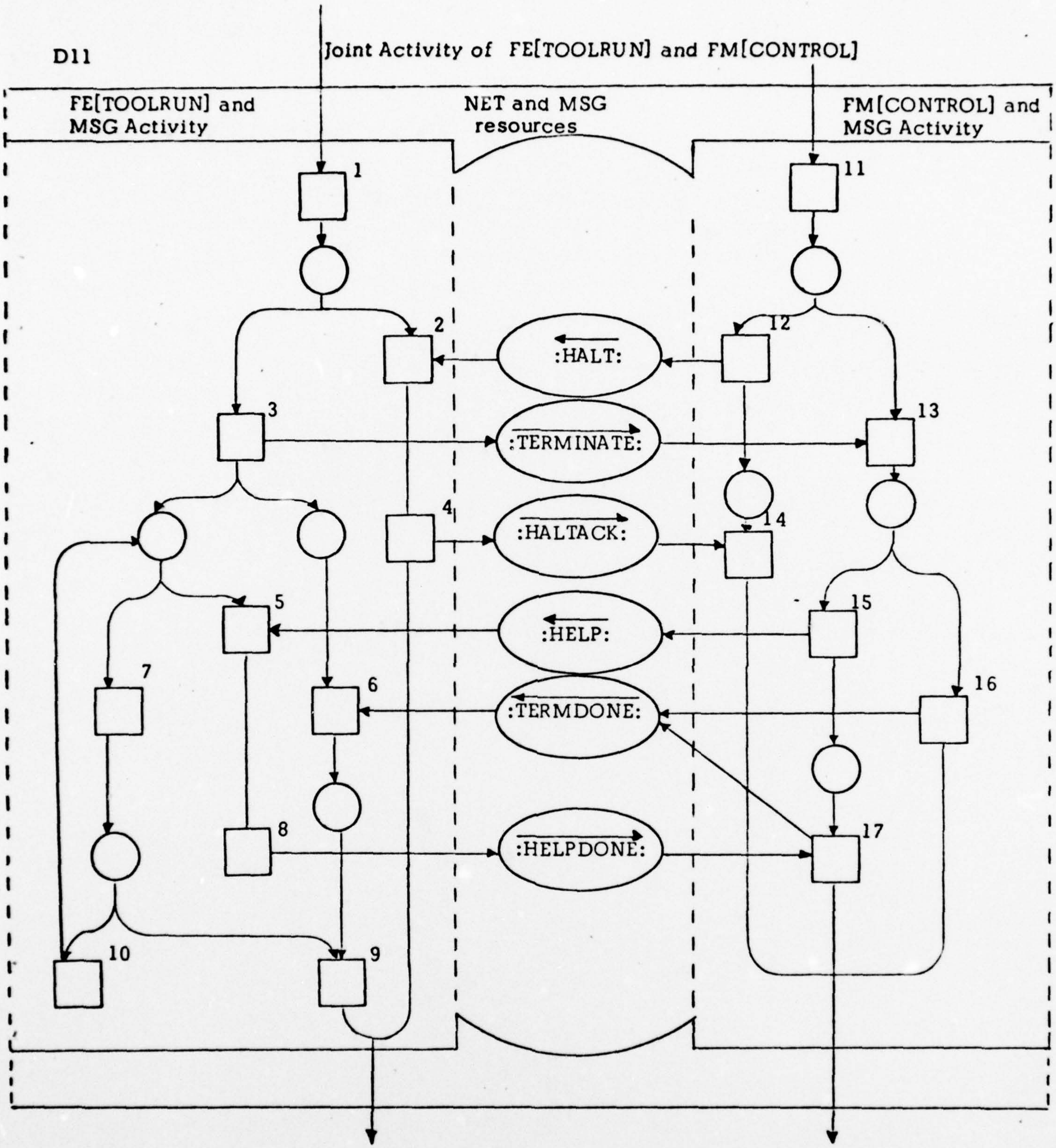
can be reduced to a single box with the inputs of 1 and the outputs of 3.

.5 Analogously, all instances of 

can be reduced to a single circle under analogous conditions, as described in .1.

With these rules and some others governing the format of the output one can produce, from D2, D4 - D8, C2, and a picture which formally expresses MSG's function as message transmitter, the following output:

D11


Joint Activity of FE[TOOLRUN] and FM[CONTROL]
FE[TOOLRUN] and MSG Activity
NET and MSG resources
FM[CONTROL] and MSG Activity
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
:HALT:
:TERMINATE:
:HALTACK:
:HELP:
:TERMDONE:
:HELPDONE:

In this diagram one new convention is used: an undirected link between two activity boxes - as between boxes 2 and 4, as also between boxes 5 and 8. The undirected link means that the two boxes so connected are representations of one-and-the-same activity - i.e. that they coincide. The same convention will apply to resource circles.

We can now subject the diagram D11 to analysis for properties of the joint behavior which it exhibits. We can ask questions such as:

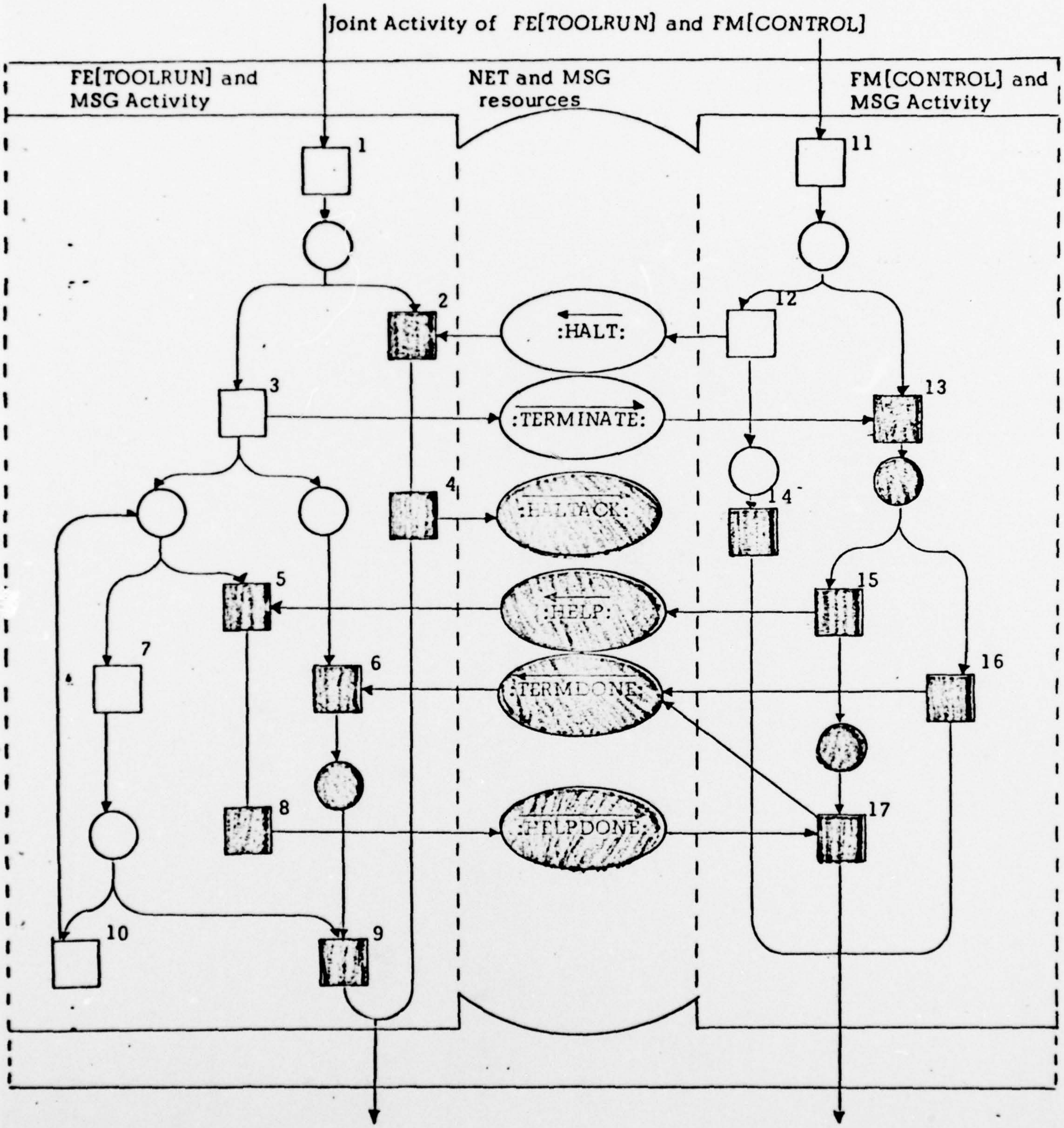
D12

.1 Given some initial conditions on the state of the resources internal to the joint activity, and assuming that the external inputs to that activity are available, will the activity take place and, as described, produce its outputs?

.2 What will be true of the state of the internal resources of the joint activity upon its completion? (E.g. may there be "left-over" messages in buffers which would have to be removed before the activity could take place again?)

Suppose we initialize the NET and MSG buffers so that all messages internal to D11 are absent, and we assume that the inputs to the joint activity are available. The unshaded portion of the next diagram represents a joint behavior which is now possible.

D13


Joint Activity of FE[TOOLRUN] and FM[CONTROL]
FE[TOOLRUN] and MSG Activity
NET and MSG resources
FM[CONTROL] and MSG Activity
1
11
2
12
:HALT:
3
:TERMINATE:
13
4
:HALTACK:
14
5
:HELP:
15
7
6
:TERMDONE:
16
8
:HELPDONE:
17
10
9

Possible behaviors are generated by resolving choices which are free within the diagram, in some particular way - in the present instance, the choice between 12 and 13 (resolved in favor of 12) and the choice between 2 and 3 (resolved in favor of 3). The resolution of these choices as shown have as consequence the guaranteed exclusion of various states and various activities which the diagram shows as possible. In the present case, all exclusions other than the exclusion of 2 and of 13, follow from these exclusions by rules which are algorithmically implementable.[1,2]

By interpretation we have displayed the following possible behavior. The FE (and HENRY) have chosen to terminate COMPILER while, concurrently COMPILER has halted. The terminate and halt messages are passed in opposite directions, but neither will be received because the FE is now waiting for a 'terminate done' message while the FM is waiting for a 'halt-acknowledge'. The FE will enter a "holding pattern", repeatedly checking for the presence of 'help' messages and for the 'terminate-done' message in its buffer.[3]

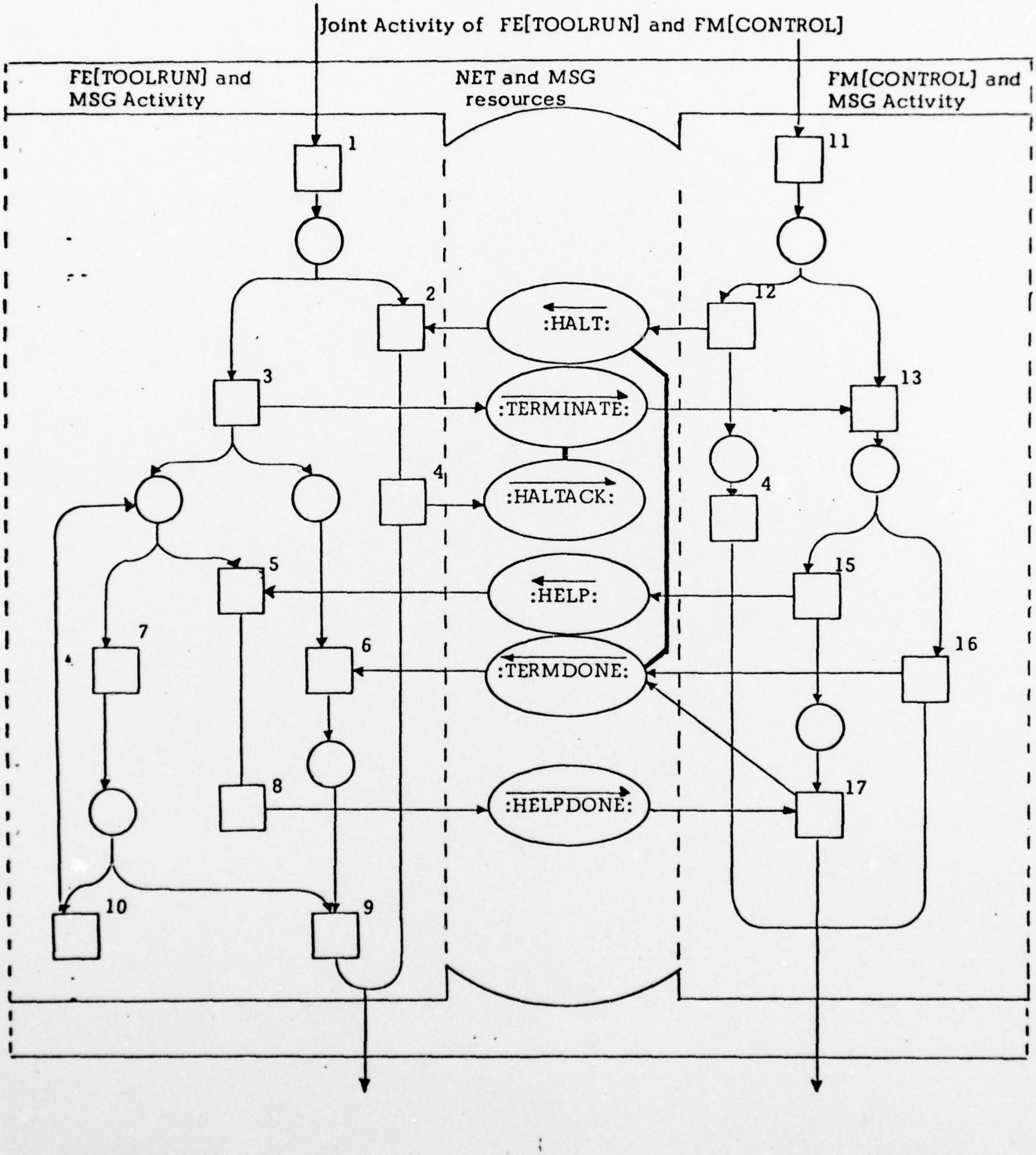
[1] The DSLP algorithm would generate the exclusions represented by the shadings by an "exclusion generation-and-propagation rule".

[2] The correctness of these exclusions depends on the assumption that neither input for the joint activity can be regenerated before both outputs of the last execution have been produced. These assumptions are justified in the present case because, by interpretation the inputs include the "control pointer" for two processes - FE[TOOLRUN] and FM[CONTROL] which are assumed to be sequential processes.

[3] The repeated checking for 'help' messages is part of the FE logic because more than one 'help' message might be sent before 'terminate-done' is sent. One can prove, however, that the joint behavior of the FE and FM could fail even if at most one help message were expected and the repeated check for 'help' messages were left out of the FE logic (left here as exercise for the reader).

The faulty design of the FE and FM activities that has just been exhibited was corrected in a manner which, when re-extracted in the manner of D9, produces the following result.

D14



Joint Activity of FE[TOOLRUN] and FM[CONTROL]
FE[TOOLRUN] and MSG Activity
NET and MSG resources
FM[CONTROL] and MSG Activity
1
2
3
4
5
6
7
8
9
10
11
12
13
4
15
16
17
:HALT:
:TERMINATE:
:HALTACK:
:HELP:
:TERMDONE:
:HELPDONE:

D14 is identical to D9 except for the identification of :$\overrightarrow{\text{TERMINATE}}$: and :$\overleftarrow{\text{HALTACK}}$: on the one hand, and the identification of :$\overrightarrow{\text{HALT}}$: and :$\overleftarrow{\text{TERMDONE}}$: on the other. The reader can verify that under the assumed initial conditions, the joint activity is now guaranteed to go through.

E The NSW Design Examples in Scenario Language

As already mentioned in section A above, Scenario Language has been used during the last year to carry out NSW design work. While Scenario Language does not directly support algorithmic procedures such as those demonstrated in section D, it has other advantages. With this language the user develops a labelled graphic presentation together with verbally presented explanations and details, indexed to the labels in the graphic part.

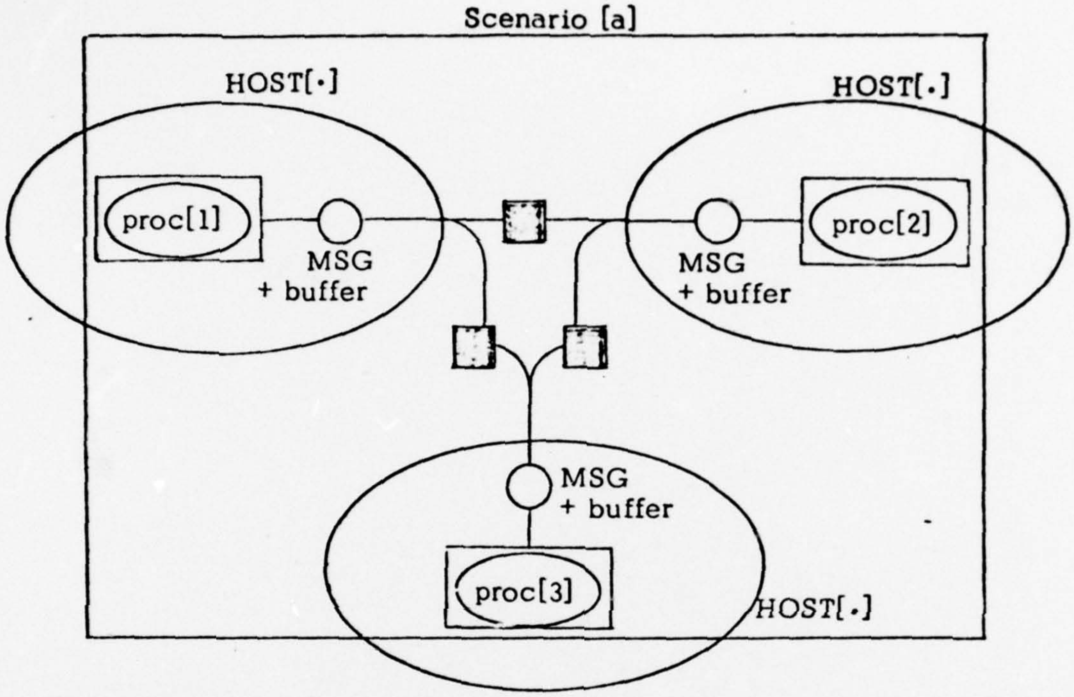
The graphic part of a scenario allows one to get a condensed overview on the communication relations between the processes concerned, and it is easy for the designer to create. With a small number of shorthand conventions it focuses the designer's attention on timing and control relations which are met in very many NSW design contexts. The Scenario Language presentations are also intended to be source documents for implementers, and they have proved successful from that point of view.

The graphic part of Scenario Language is easy to explain in terms of DL1, as will be shown below. This justifies the expectation that useful forms of expression (such as Scenario Language) may be formally translatable into DL1. Thus system designers, when the need arises, could express themselves in various special forms without giving up the services which DT1 is expected to render. Indeed, there is reason to expect that DL1 expressions produced from Scenario Language expression by translation would be more tractable to checking algorithms than DL1 expressions without such a special origin.

We shall now explain most of conventions which govern the graphics part of Scenario Language in terms of DL1.

A scenario is an activity carried on by a set of intercommunicating processes which jointly accomplish some system function - e.g. login, toolhalt, filedeliver, etc. Each of these processes is internal to some NSW host, and no two of them are assumed to be internal to the same host (although they are allowed to be resident on the same host). A typical scenario might have the following form.

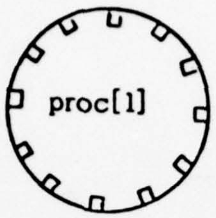
E1



A scenario with three processes

The shaded boxes represent inter- or intra-host message transfers, just as in C10.

Scenario processes are represented in the form of a clock face.



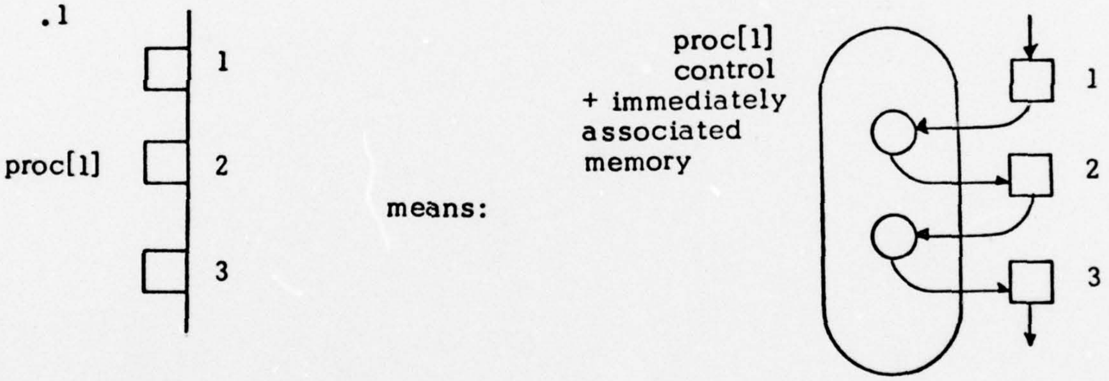
The 12 boxes are 12 available symbols for activities of proc[1] . This is adequate for all processes with 12 or fewer activities. If the process description requires 24 or fewer activities, a clock face is used with 24 boxes by interpolating a box between each pair of boxes occupying the standard clock face positions (more than 24 have not yet been required).

We shall now describe the way in which process control over these activities is expressed.

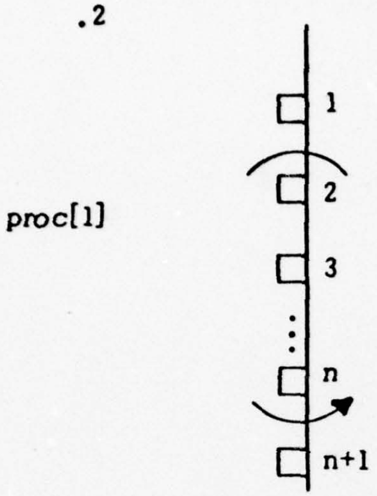
In general, control progresses from activity to activity around the clock face in a clockwise direction. The clock face segments we shall represent below are to be read from top to bottom.

E2

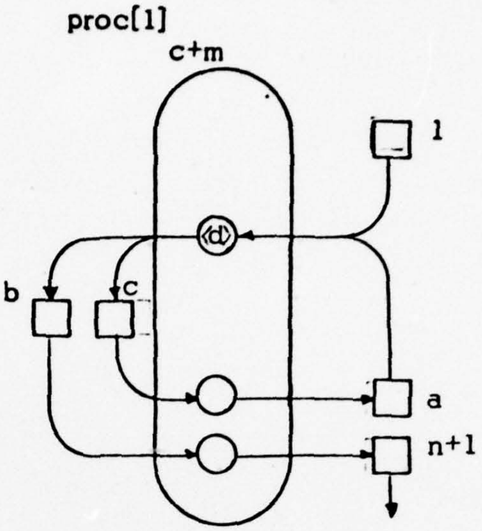
.1



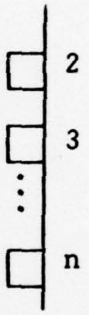
.2



means:

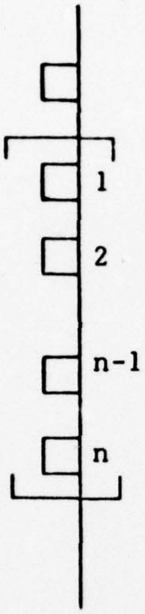


where a is the activity

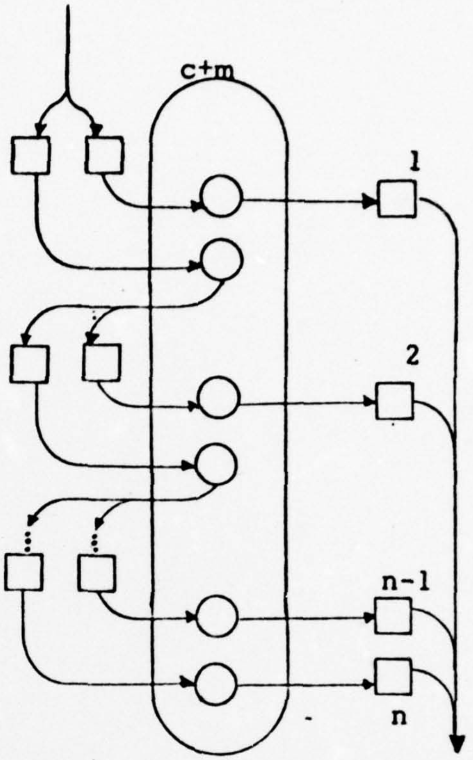


State d is followed by a test on something external or internal to proc[1] which decides whether activity a next takes place or not. Thus, after activity 1, activity a will take place zero-or-more times, and then activity n+1 will take place.

.3

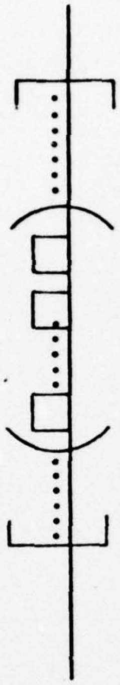


means:

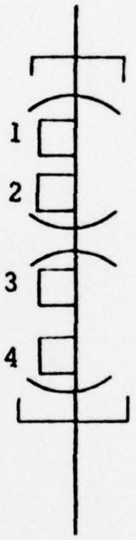


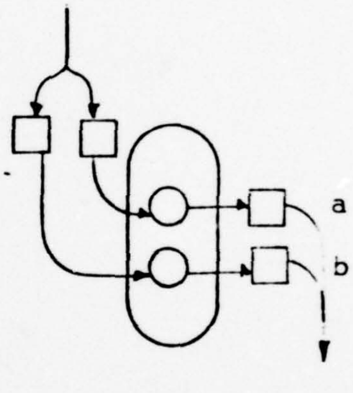
One-and-only one of the activies in the square brackets is to occur, depending on conditions to be tested for in the indicated sequence.

.4

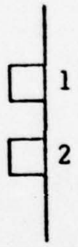


a parenthesized sequence of activity boxes inside of a square bracket represents a single compound activity whose possibility of occurrence is tested for. For example:

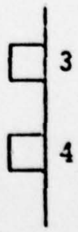




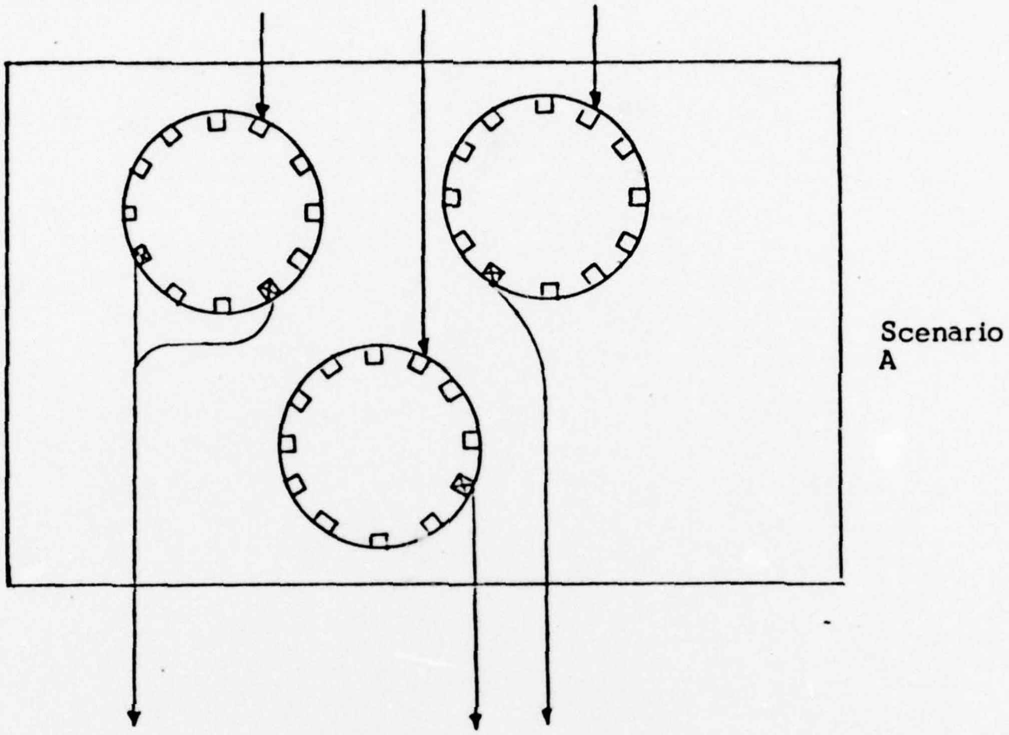
where activity a is



and activity b is

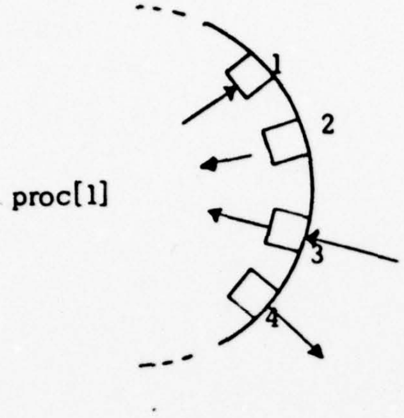


E3 By convention, the initial activity of each process which is part of a scenario is represented by the box at 1 o'clock on the clock face. Terminal activity boxes are marked by an 'x'.

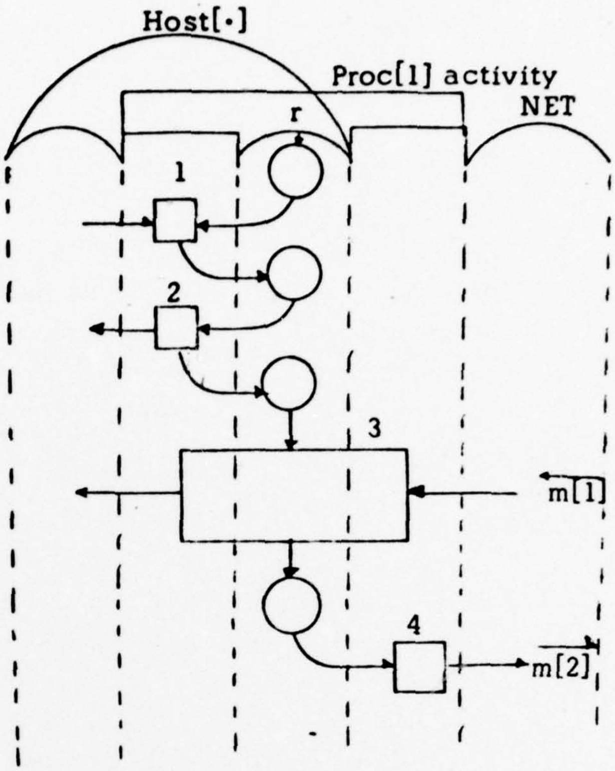


E4 Information flow

One-or-more arrows may lead into or out of each box of a clockface. They represent information flow: if the body of the arrow lies on the inside of the clockface circle it represents information flow between the process and some other entity in the same host as the process - including, or course, resources internal to the process - while arrows whose body lies outside of the circle represent message flow between the process and other processes that are part of the same scenario (or possibly other scenarios). Thus:



implies:



r: c+m of proc[1] + the necessary parts of MSG

E5 Process identification

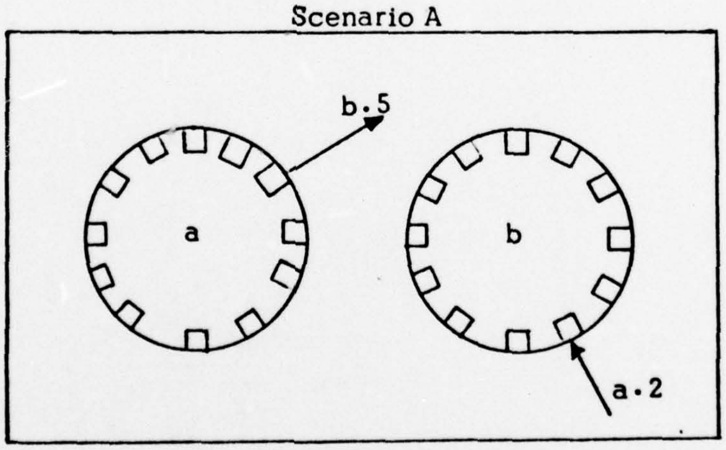
In NSW there are the Front-End FE, Works Manager WM, Foreman FM, and File Package FP capabilities, and every process which is part of a scenario belongs to one of them. A given process of a given scenario can usually be identified by the capability to which it belongs. For example: the WM's participation in LOGIN defines a unique process; the FM's participation in TOOLRUN does likewise; etc. (Occassionally this fails as in the case of host-to-host file transfers. FP participates in this both as dispatcher and as receiver.) In the dominant case therefore it is enough to write the appropriate NSW capability label at the center of a clockface to identify the process uniquely within the given scenario.

Sometimes a clockface in a scenario represents the detail of a single activity box in some other named process within a given capability. For example: HALT is a scenario. It contains an FM process (represented by a clockface) which details an activity called HALT of another particular FM process called FM[CONTROL]. The FM HALT-clockface will show the label FM[CONTROL] instead of just FM.

E6 Message addresses

Outside arrows - i.e. those whose body lies outside of a clockface circle - always have message addresses and some indication of *message content attached. The conventions are as follows:*

.1

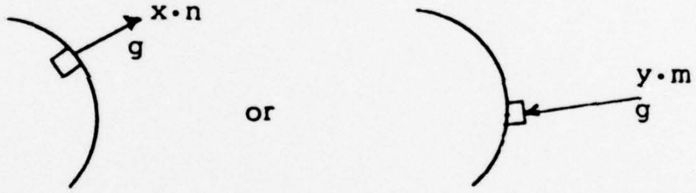


b.5: a message sent by process a (of Scenario A) as an output of the activity represented at clock position 2, which is intended as an input to the activity represented at clock position 5 of process b . Thus, the source address of this message is a , and the destination address b . To actually make the message an input to activity 5 may be implemented in many different ways: the timing relations between a and b may guarantee that when b is ready for activity 5, the only message from a which could possibly be in the buffer is b·5 ; the message, as part of its content may contain additional identifiers which make it recognizable as the input to activity 5; etc.

a.2: a message produced by process a as an output of activity 2 which is an input to activity 5 (of process b). Thus 'a.2' and 'b·5' are two different names for the same message.

.2 Generic vs. specific addresses

Message addresses in NSW may be <u>generic</u> or <u>specific</u> (see C7 above). In Scenario Language the letter 'g' is used to indicate a message with a generic address

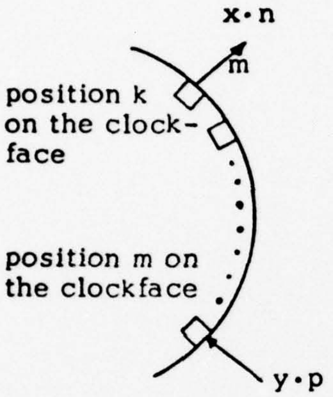


The absence of the letter 'g' is taken to mean that the address is <u>specific</u>.

E7 Timers

Whenever a process sends a message over the net for which it expects a response, it sets a timer. Failure of the response to arrive will be registered in the form of a local message from the timer that the time limit has passed. In Scenario Language the use of such a timer is indicated by a single number associated with the arrow of the message requiring a response, thus:

.1



y: the process from which a response to message x·n is expected (usually y=x)

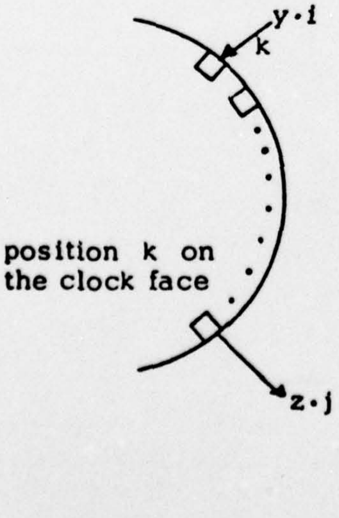
m: the clockface position where the activity which requires the timely response from y is represented

(If m = k+1 the process can "go to sleep" while waiting for the response (as happens twice in D4); otherwise it functions as with the first timer in D8 which is set to wait for the response fm:DONE .)

In any case the activity in clock position m of .1 requires as input the message y·p with "timer blessing" - i.e. verified as having arrived in time. *If this activity cannot take place because* the response does not come in time some other activity takes place next which may or may not be explicitly exhibited on the clock face. In any case reference to the verbal part of the scenario presentation is required to determine how the process continues. The clockfaces are there, in the main, to focus attention on the "normal case".

Finally, a process also sets a timer when it receives a message to which it is supposed to respond.

.2

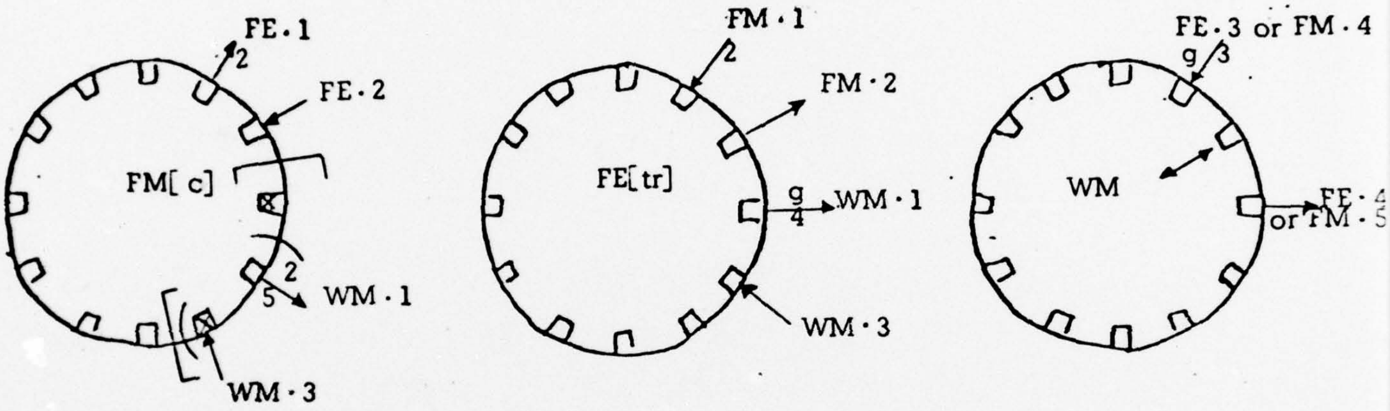


(usually y=z)

If the timer passes its limit before the response is ready, y sends a substitute message to z which says in effect "I'm working on it". The timer is then re-set so that another such time marker may be sent if the response is still not ready. This mechanism has no other explicit reflection in the graphic part of Scenario Language, and we left it totally out of account in section D.

E8 The two Scenario examples below include four of the processes shown in section D.

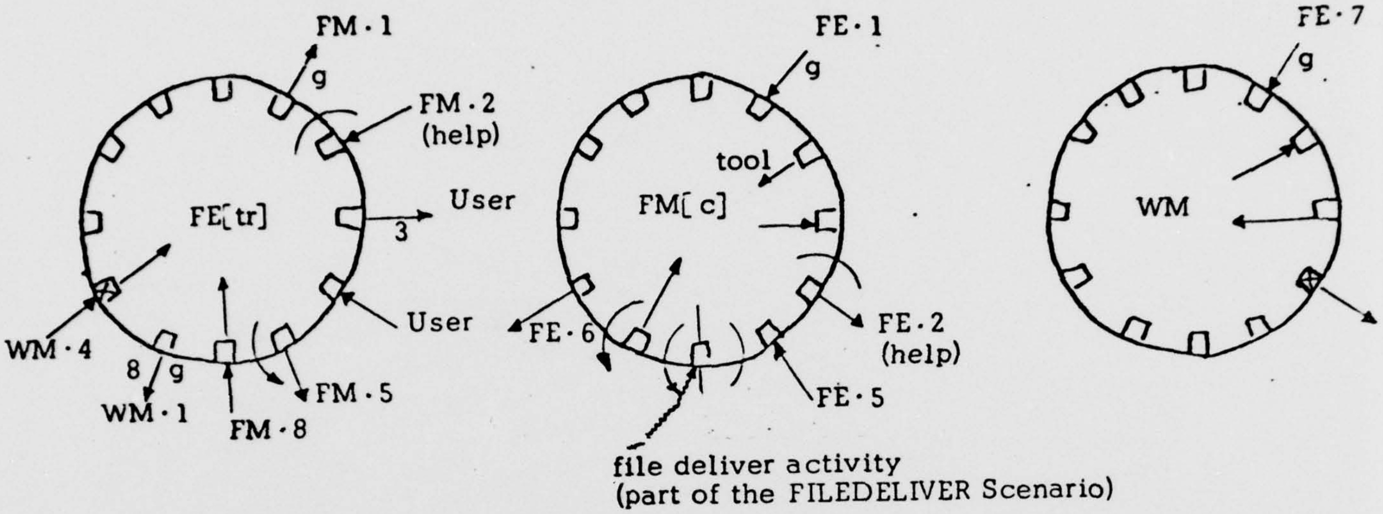
.1 The HALT Scenario



c: [CONTROL]
tr: [TOOLRUN]

Compare FM[c] to D4 and FE[tr] to D7.

.2 The TERMINATE Scenario



Compare FE[tr] to D5 and FM[c] to D8.

III. Choice and Cause: A Formal Analysis Based on Petri-nets

III. Choice and Cause: A Formal Analysis Based on Petri-nets

A. Introduction

This is a work of theory construction on behalf of practical objectives in the general domain of systems analysis. Systems, as meant here, are to be understood as spatially distributed domains of formally organized purposeful activity - of men, or machines, or a mix of the two, which is the general case. What forms such domains into meaningful wholes is the coordination of the actions and states of many participants - mechanical or human - by means of communication. We have, therefore, also used the phrase domains of formal communication as a short pointer **at what** 'system' means to us.

This notion of 'system' does not immediately focus attention on input/output relations, but more immediately on the definition of dynamic (communicative) relations between parts. The stock exchange, or the U.S. mail service are natural candidates for being viewed as systems, but their important functional features are not captured by talking about their "inputs" and "outputs". We, of course, do not mean that the notions "input" and "output" are irrelevant to systems - we merely mean that they do not occupy as central a position in our scheme of things as they do in most other approaches to systems thinking.

The theoretical work is intended to support the deduction of (causal) dependence relations between presences of state or action at various times and places from structural descriptions of systems - for example, to declare interdependencies among the presences and contents of messages distributed over (system) space and (system) time, from a set of formally defined communication protocols. Deducing the relations between the content of input and the content of output from given rules of calculation also lies within the class of analytic problems alluded to above. This class of analytic problems is immense. in general, immensely complex, and always of great importance when vital business or government functions depend on systems involving technologically implemented data processing and communication. These facts justify considerable effort in the development of theoretical support for such analyses.

This paper focuses on a technical notion of 'choice' and its relation to 'cause'. Choice and cause have not been generally seen as natural bedfellows. Rather, 'cause' has been associated with the idea of necessary temporal precedence. We shall, of course, not dispute the importance of that association. But, as we see it, necessary temporal precedence is not coterminous with cause, as the following example suggests.

A1 An executive, in the performance of his daily duties, always makes choice A and then choice B. Choice A has the possible results yes (A), no (A); choice B has the possible results yes (B), no (B). There are thus, for each day, four thinkable histories: yes (A) → yes (B) ... etc. Now note:

.1 The fact alone that on every day yes/no (A) precedes yes/no (B) does not mean that the resolution of choice A exerts a causal influence on the resolution of choice B.

.2 The fact alone that in the given system context all four histories are possible does not mean that the resolution of choice A exerts no causal influence on the resolution of choice B.[1]

Consider a segment of organizational history in which yes (A) precedes yes (B). Because the same executive participates in both decisions, yes (A) <u>necessarily</u> precedes yes (B), but: does yes (A), wholly or partially, <u>cause</u> yes (B)? Not necessarily. However, yes (A) could not wholly or partially cause yes (B) unless it were temporally prior. So much for the example.

[1] Suppose that, in every execution of a repeated program cycle we compute B := A $\oplus$ C, where A, B and C are binary variables and '$\oplus$' is binary add. Although the next value of B is <u>influenced</u> by the last value of A, all four combinations of A argument value and B result value are possible, if C is independent of A.

The contributions to the subject of choice and its relation to cause in this work are partly technical (executed with the help of Petri-net concepts) and partly general philosophic. The latter aspect of the work has, we believe, uses in its own right in helping to focus the attention of those interested in the theory and practice of systems analysis on issues of importance, approached by no matter what technical means.

B. Choice Structure Broadly Characterized

The word "choice" as commonly employed harbors some ambiguities and general meaning tendencies from which we shall have to free ourselves.

First of all, "choice" is used to mean a set of alternatives which present themselves to someone from among which he may choose - as on a menu, choice between steak and lobster. "Choice" is also used to mean the outcome of choosing - e.g. 'steak is my choice'. We shall use choice in the former sense, and refer to the latter as choice result, or choice outcome. We shall refer to the process of arriving at the result as choice resolution.

Secondly, there is the ambiguity as to whether one is talking about a choice on a given occasion, or thinking of a given choice as something which may arise repeatedly on many occasions. We shall always mean the latter. Thus, associated with a given choice - or choice situation - there is to be the possibility of repeated occurrence of choice resolution.

While a choice is most often thought of as something which confronts a person, we shall also apply the idea to physical situations in which personal choosing is not directly involved. For example, a ball balanced on a knife blade may drop to the right or drop to the left. Its dropping to one side or the other may be viewed as the resolution of a choice.

We shall always view choices - or choice situations - as structures associated with localities in a spatially extensive domain of activity. Influences which wholly or partially determine particular choice resolutions may flow to the locality where the choice is situated - either from other parts of the system of activity, or from some system-external source. Suppose, for instance, that in some locality there are two hoppers into which items are sorted. The deposit of each item in one or the other hopper involves the resolution of a choice; but in a given system context, the resolution of that choice on each occasion may be wholly determined by the surrounding system. Thus, the presence of choice in some locality does not ipso facto mean the presence of system indeterminacy; it only means the local existence of alternative outcome possibilities. Indeterminacy exists only if the choice is subject to influences which lie beyond the boundary of the system of activity of which the choice is a part.

We can conveniently divide the subject matter of choice-structure into micro-structure and macro-structure. Micro-structure deals with the description of a locality where a choice is resident; macro-structure deals with the description of how, in a systems context, such localities relate to one another. Most of this paper deals with the micro-structural aspects of choices. The micro-structure of a choice involves:

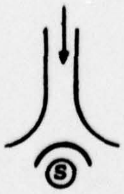
B1 .1 A set of elements in terms of which the determinations governing choice resolutions are expressed. These elements are called the determinants of the choice. [I.e. determinants are used to express determination.]

.2 A set of elements in terms of which the results of choosing are expressed. These elements are called the resultants of the choice. [I.e. resultants are used to express results.]

.3 Structural connections between the determinants and the resultants which, in effect, define the mapping from choice determination to choice result.

B2 An Example:

A choice is defined for a fork in a road, in the context of a system in which controlled motion along the road towards the fork and beyond the fork are defined activities.



Assume that at the fork there is a signal which may be set to point to the left or point to the right with the intention of directing travelers to advance in one or the other direction.

Two possible outcomes of the defined choice are envisaged which we may call going right and going left. We shall call each of these a resultant of the choice.

The outcome going right is to depend on the prior presence of a traveler on the approach to the fork and the signal set to right; the outcome going left is to depend upon the prior presence of a traveler on the approach, and the signal set to left. We shall take traveler, right and left as designations for three determinants of the choice.

The structural connections between determinants and resultants should now assure the following: that the co-presence of traveler and right results in an occurrence of going right, while the co-presence of traveler and left yields an occurrence of going left.

We can more accurately describe the two results thus: an occurrence of going right and a co-exclusion (from occurrence) of going left (and vice versa).

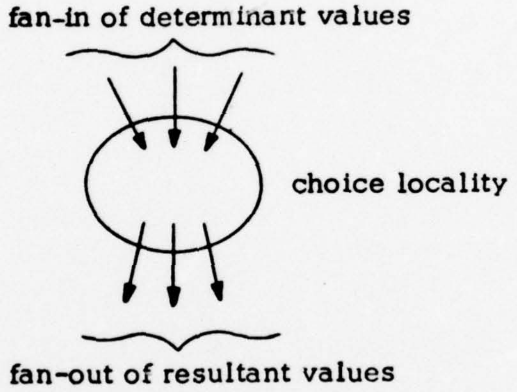
Similarly, more accurate descriptions of the two choice determining conditions are: the co-presence of traveler and right, and the co-absence of left - and similarly with right and left interchanged. This form helps to express the idea that, for a "clean" choice result, the signal must be in an unambiguous state.

Thus, we see that choice determinations may be expressed in terms of some pattern of occurrence and exclusion of the choice determinants - and similarly for the expression of choice results in terms of its resultants.

The possibility of an ambiguous signal state at a time when a choice must be resolved is, in our account, not only a possibility to be reckoned with in the physical implementation of systems, but finds its expression in the theoretical structures we build to represent choices and their interrelations. The micro-structure of choices will always admit the possibility of incoherent choice determinations, and incoherent choice results. The macro-structure may insure that incoherent determinations and results do not arise, but that is generally the subject of analysis and proof. More will be said about coherence and incoherence in the sequel.

The general reason that choice determinations and choice results are expressed in terms of a multitude of determinants and resultants is that each choice resolution is generally influenced by a multitude of sources and, in its turn, influences a multitude of destinations. The influences are "fanned in" to the choice locality from neighboring localities, and the result is "fanned out" to other neighboring localities.

B3



For example:

- I buy a pack of Marlborough cigarettes at a vending machine

  <u>fan in:</u> me with my preference; the vending machine with its supply

  <u>fan out:</u> me with Marlboroughs; the machine with reduced Marlborough supply

- The vending machine is re-supplied

  influenced by my purchase

- I offer Marlboroughs to a friend

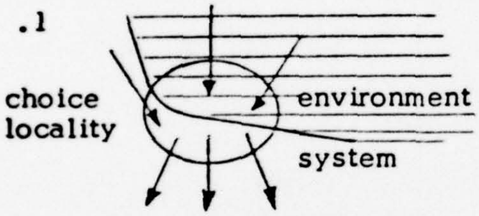
  also influenced by my purchase, but at a different location

Local choices connect to one another to form a macro-structure since the determinants of a choice are the resultants of other choices. The effect of choice resolution is to "assign value" to the determinants of neighboring choices, and thus exert an influence on their next resolution. With reference to B2, the traveler arrives at the locality as a result of a prior and spatially adjacent choice resolution, and so also does the next setting of the signal to right or left "arrive" on site as a result of prior and spatially adjacent choice resolution. The effect of going right and not going left (or vice versa) will be understood as exerting an influence on subsequent and adjacent choices.

A choice locality may be at the boundary of a system or in its interior. If it is at the boundary then some of its determinant values may be supplied by the system environment and/or some of its resultants may be supplied to the environment.

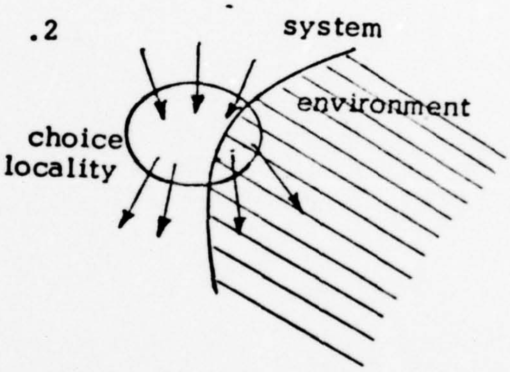
B4

.1



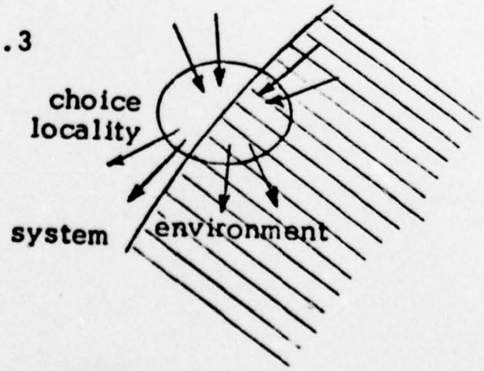
The environment supplies some of the choice determinants

.2



The environment receives some of the choice resultants

.3



The environment both supplies some of the determinants and receives some of the resultants

Example: An executive in whose discretion it lies to make a choice, partially constrained by system-defined conditions, participates in a choice which lies at the boundary - i.e. a choice with determinants which are not system-defined.

Environmental influence (via determinant values) on a system-defined choice is our model of system input; the system exerting an influence on an environment-defined choice is our model of system output. In the case B4.3, we see a choice which is both system-defined and environment-defined. (For a particularly important class of system models, every system choice which lies at the boundary will be "cut" in the manner shown in B4.3.)

The question now arises: have we not, with unconventional terminology, described structures very similar to, or identical with, structures that are well-known to systems engineers and analysts? Suppose we thought of each local choice as describable by a locally implemented function which maps a vector of determinant values to a vector of resultant values. Each choice point is then a function node connected to other function nodes by lines thought of as value variables carrying values from node to node. Are not models which use these last-named concepts rich enough to represent adequately the system relations which we are striving to model? Yes and no.

First of all, value variables are assumed to range over some finite set of possible mutually exclusive values - e.g. bit values, 0 or 1. On the other hand, choice determinants and resultants, when determined, are determined as present or absent - occurring, or excluded from occurring. In standard system models, in which function nodes are interconnected by 'data lines', one usually also considers another level of structure, namely, control: function evaluations are enabled by the arrival of control signals which, at any given time, may be present or absent - like (but not exactly like) the determinants of choices.

In logic design, on the other hand, where one also deals with interconnected function nodes, one needs, in addition, delay elements which create relations between presences and absences (of changes). Broadly speaking, one can say that in various existing disciplines of system thinking, function and timing are not conceptually united. (This follows, naturally, from the fact that the abstraction 'function' as used in mathematics is in no way directly related to cause and, more generally, temporal relation.) In contra-distinction, one may say that our approach (choice structure, in its micro- and macro-aspects) aims to deal with function and timing in a wholly uniform manner. Presence and absence at some time (suitably formalized) is taken as primitive, and data-values which enter or exit from a computation are constructs over the primitives. In the end effect, function and timing are not separable, since, as is well known, ill-timings will produce ill-functions in the most bizarre and unpredictable ways. These difficulties arise in part because the mental trick of separating function from timing is not faithful to implemented system reality.

If we are successful, we shall develop, from the beginnings described below, a class of system models which by virtue of the formal (axiomatic) constraints imposed, will guarantee a higher fidelity in the representation of implementable system realities than methods now in existence. While the notion 'function' has nothing directly to do with implementable process, the notion choice (which, in our context, stands in the place of 'function') does.

In sum: we said 'yes and no' above for the following reason. On the one hand, standard modeling techniques might be viewed as too powerful, in that they allow one to discourse with equal facility about unicorns (which don't exist) as about horses (which do). On the other hand, standard techniques are weak in representing loosely coupled systems (e.g. communications networks, rather than clocked computers).

We believe in the existence of a technical conceptual framework for the description and analysis of spatially distributed domains of formally organized, purposeful activity -- of men, or machines, or a mix of the two, which is the general case. System diseases, such as deadlock, critical race, instabilities of various kinds, etc. are endemic to every setting which conforms to our description, be it a formally defined game of strategy in which men play with counters on a board, or a pilot steering a craft with the aid of avionics. The work on choice and cause is, we believe, a contribution to the construction of that technical conceptual framework.

C. Petri-net Preliminaries to Choice Structure

We shall base our technical constructions on Petri-nets. Petri-nets lend themselves to the description of the patterns of interaction between a multitude of organizational elements, whose individual behavior can be characterized in terms of state transition diagrams. In this work, no direct reference will be made to the parsing of total system behavior into the behavior of such elements. All that will be important is that total system states are representable in terms of a multitude of state elements, and that total system state changes are representable in terms of a multitude of event elements. These multitudes reflect the fact that the domain of activity is assumed to be spatially extensive: the total state is decomposed into many elements which, in various combinations, describe what is true in many localities. The total change is decomposed into event elements which, in various combinations, describe changes at various localities. In the following formal definition of Petri-nets, we follow C.A. Petri, their original inventor.

C1 We shall take Petri-nets η to be defined thus: $\eta = \langle S,E,F \rangle$ where S is a set of state elements, E is a set of event elements and $F \subseteq S \times E \cup E \times S$, the "flow relation" represented by directed arcs in the graphic representation of η. The elementary axioms are:

.1 $S \cup E \neq \emptyset$

.2 $S \cap E = \emptyset$

.3 domain (F) $\cup$ range (F) = $S \cup E$

- We will designate by X , the set of all net elements

$$X = S \cup E$$

- $\forall x \in X$: $\quad x^\bullet \overset{\Delta}{=} \{y \in X:\ xFy\}$; $\quad x^\bullet$ is called the <u>post-set</u> of x

- Similarly $\quad {}^\bullet x \overset{\Delta}{=} \{y \in X:\ yFx\}$; $\quad {}^\bullet x$ is called the <u>pre-set</u> of x

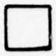
- $\forall A \subseteq X$: $\quad A^\bullet \overset{\Delta}{=} \bigcup\limits_{x \in A} x^\bullet$; $\quad A^\bullet$ is the <u>post-set</u> of A

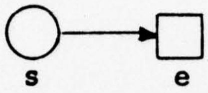
- $\forall A \subseteq X$: $\quad {}^\bullet A \overset{\Delta}{=} \bigcup\limits_{x \in A} {}^\bullet x$; $\quad {}^\bullet A$ is the <u>pre-set</u> of A

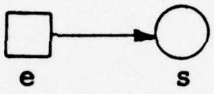
Comment: C1.3 means that $\forall x \in X$: $x^\bullet \cup {}^\bullet x \neq \emptyset$

Petri-nets are graphically represented in the form of bipartite graphs: one vertex type for state elements and a second vertex type for event elements.

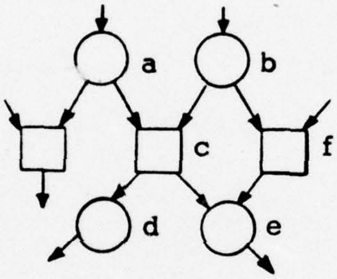
C2 .1 State elements:

.2 Event elements: 

.3 $\langle s,e \rangle \in F$: 

.4 $\langle e,s \rangle \in F$: 

.5 ${}^{\bullet}c = \{a,b\}$
$c^{\bullet} = \{d,e\}$
${}^{\bullet}e = \{c,f\}$



We shall now discuss, rather superficially, the main features of Petri-net interpretation which serve as a point of departure for the work on choice structure.

C3 We can think of the events as events of production which consume a set of inputs and produce a set of outputs. The state elements ${}^{\bullet}e$ are then the ready-states of the e inputs and $e^{\bullet}$ are the ready-states (this time, 'ready' in the sense of "done") of the e outputs. Then $|{}^{\bullet}e|$ (the number of elements in ${}^{\bullet}e$) is the number of inputs e requires, and $|e^{\bullet}|$ is the number of outputs that e produces.

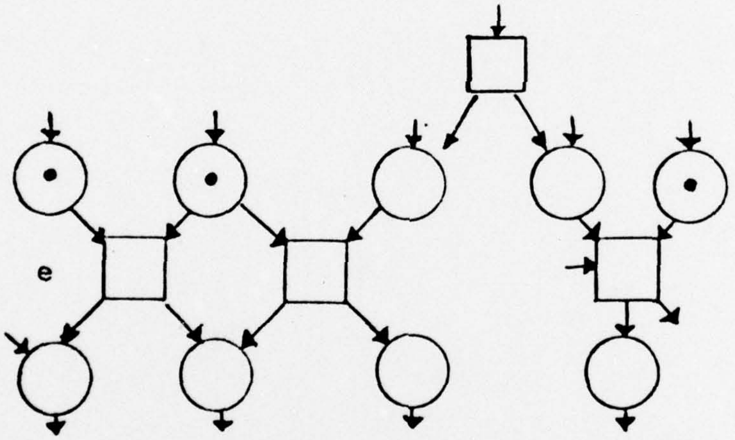
C4 An event can occur if all of its inputs are ready - i.e. the state $\bigwedge_{s \in {}^{\bullet}e} s$ holds. The result of an event occurrence is that $\bigwedge_{s \in e^{\bullet}} s$ holds.

C5 Correspondingly, one may represent system states on a Petri-net graph by distribution of markers -- or tokens -- over the state vertices of the graph. Such distributions are called markings -- or more exactly state markings -- of a net: each marked state is asserted to hold as part of the represented system state, while each unmarked state element is asserted not to hold as part of the current system state. As per C3.1, the individual markers, or lacks of them, viewed as elements of state, assert the presence or absence of inputs (outputs) of event elements. Formally, we shall take markings to be subsets of state elements: those and only those state elements belonging to the subset are asserted to hold.

C6 Local changes of state are represented by the "firing" of event elements, according to the rule implied by C4: if, at a marking m , $^\bullet e \subseteq m$ (i.e. all input - ready states - of e are marked) then the event element is enabled and a change can take place -- a change that is representable by removing the markers from all inputs of e , and placing markers on all outputs of e .

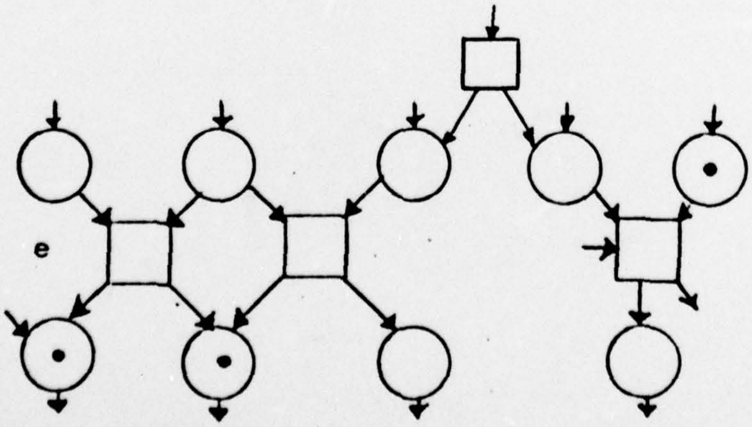
C7

1.



a marking at which e is firable.

2.



the new marking which results from the firing of e.

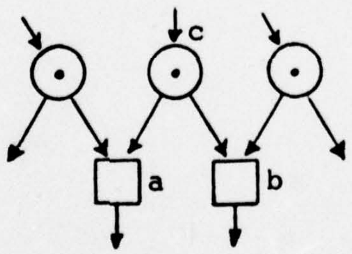
A marking class is a family of markings all related to one another by event firings. Marking classes are, by interpretation, the set of all possible well-formed system states - exclusive, say, of states which may result from component failures, illegal inputs, and other classes of misbehavior.

C8 Several important questions now arise about the very simple "firing rule" stated above. The first is this.

.1 Are markings m with the property that, for some event e, $^\bullet e \subseteq m$ and $e^\bullet \cap m \neq \emptyset$ to be viewed as legitimate members of marking classes? The enabling condition, $^\bullet e \subseteq m$ allows us to "fire" e, but how is one to place a marker on an output which already has a marker? The presence of two markers on a state element is meaningless under our general scheme of interpretation; one cannot "doubly assert" that a state element holds. Marking classes M with the property: $\forall m \in M: \forall e \in E: {}^\bullet e \subseteq m \Rightarrow e^\bullet \cap m = \emptyset$ are called "safe" and many workers assume that only safe marking classes can be used to define the well-behavior of systems. Other workers, notably C.A. Petri in his current work, require that the "basic" firing rule make a stronger demand for event enabling, namely: $^\bullet e \subseteq m$ and $e^\bullet \cap m = \emptyset$. This work on choice follows the former approach to the definition of event enabling.

.2 A second question of importance about the firing rule is this: How is it to be applied at a marking m in which, two events, e_1 and e_2 are enabled - i.e. $^\bullet e_1 \cup {}^\bullet e_2 \subseteq m$ and $^\bullet e_1 \cap {}^\bullet e_2 \neq \emptyset$.

Example:



With the interpretation in mind that inputs to events represent indivisible resources which are consumed when the event occurs, one views the two events a and b as in conflict with one another over resource c . Although, at the marking shown, both events are enabled, an occurrence of either of them destroys the pre-conditions which enable the other. This, therefore, is generally interpreted as a choice situation: event a occurs, or alternatively event b occurs, but not both. Nothing in the net represents influences which govern the resolution of the choice.

This very superficial discussion of Petri-net interpretation serves to establish a base line to which we can refer in our development of the formal and interpreted aspects of choice structure. As we shall see, this development entails, among other things, a new view of markings and of the firing rule.

We shall give formal definition to choice structure in terms of Petri-nets. The net elements will be interpreted as the determinants and resultants of (interconnected) choices. The F-relation will be interpreted as defining the connections between the determinants and resultants of choices. In terms of these connections we shall then be able to define how choice determinations produce choice results. More fully (and more formally) we can state our program thus:

C9 .1 We shall formally define choice structure over Petri-nets. This will include the definition of the D-set, the R-set, and the interconnection between them (as per B1 , where the D-set was called the set of determinants and the R-set was called the set of resultants).

.2 We shall then explain how determinations are expressed in terms of D-sets, and how results are expressed in terms of R-sets. We may think of a choice determination as a possible marking on the D-set of a choice, and the result as a possible marking of the R-set of a choice.

.3 We shall then define a relation between determinations of a choice and results of that same choice - a relation called the compatibility relation. The compatibility relation determines what results of a choice are compatible with a given choice determination - and conversely, what determinations are compatible with a given result. We shall then have the basis for a new token game defined over nets with choice structure. The compatibility relation is, in general, many-many because system-defined choices may lie at the boundary of the system, so that the system-defined determinants only partially constrain the choice result, or the system-defined result only partially constrains the determinations which can produce it, or both. If the choice is interior to the system, then the compatibility relation will be one-one -- i.e. each choice determination will be compatible with exactly one result (and, in that sense, determine the result), and each result will be compatible with exactly one determination. To give the compatibility relation exact formal meaning, one must be able to recognize where the boundary lies. We shall have to add something to Petri-net definitions for that purpose.

.4 Finally, we shall begin (but not finish) the description of coherence properties of choice determinations and choice results -- i.e. to state general axioms which distinguish choice determinations that can lead to no resolutions from those which can (and similarly for results).

Several interpreted circumstances are formally addressed under the heading of coherence properties - e.g., in traffic systems, circumstances which lead to collisions: Collision avoidance depends upon system relations which guarantee coherent choice determinations at every point where roads merge and vehicles may reach the merge point from several directions. A problem of coherent choice determination also arises at a road branch point, as already mentioned above in connection with example B2.

According to this program, the next following sections are labeled:

- Choice Structure over Petri-nets
- Determinations and Results
- System Boundary
- The Compatibility Relation
- Coherence Properties

D. Choice Structure Over Petri-nets

We shall define a choice structure relative to a given Petri-net $\eta = \langle S, E, F \rangle$. A choice will be formally defined as a special type of subnet of η[1], and a choice structure $\mathbf{C}$ as a covering of η by subnets C of this special type.

We can describe the motivation for our formalization with the following image.

D1 The micro structure of a choice looks like a graph whose nodes are divided into two disjoint, non-empty sets - sources, and sinks. We might think of the sources as representing a set of producers and the sinks as representing a set of consumers. The graph arrows show which producers supply which consumers. The producer set and the consumer set are to be understood as relating to each other in the following special way: the consumers are all the consumers supplied by the given set of producers; and also: the producers are all the producers who supply the given set of consumers. In the case of choices, the "producers" are the choice determinants and the "consumers" are the resultants.

In a Petri-net which support choice structure (as not every Petri-net does), it is possible to partition its arcs by a set of subgraphs, each of which conforms to the model just explained.

D2 Notation

We shall subscript everything that pertains to a choice with its name, e.g.:

$$C = \langle S_C, E_C, F_C \rangle \ , \ S_C \cup E_C = X_C \qquad \text{etc.}$$

D3 Axioms about families of subnets $\mathbf{C}$

$$\forall C \in \mathbf{C} :$$

[1] $\eta' = \langle S', E', F' \rangle$ is a subnet of $\eta = \langle S, E, F \rangle$ if (a) η' is a net; (b) $S' \subseteq S$ and $E' \subseteq E$ and $F' = F \mid S' \cup E'$ (F restricted to $S' \cup E'$).

.1 $\mathrm{dom}(F_C) = S_C$ <u>or</u> $\mathrm{dom}(F_C) = E_C$

.2 $\{\mathrm{dom}(F_C)\}_{C \in \mathcal{C}}$ is a partition of $\mathrm{dom}(F)$

.3 $\{\mathit{range}(F_C)\}_{C \in \mathcal{C}}$ is a partition of $\mathrm{range}(F)$

.4 $\{F_C\}_{C \in \mathcal{C}}$ is a partition of F

D4 Definition of the D-set D_C and R-set R_C of C

.1 $D_C \overset{\Delta}{=} \mathrm{dom}(F_C)$

.2 $R_C \overset{\Delta}{=} \mathrm{range}(F_C)$

and thus

.3 in the subnet C: $D_C^{\bullet} = R_C$ and $D_C = {}^{\bullet}R_C$

D5 The following are consequences of our axioms and definitions.

.1 $D_C \subseteq S_C$ iff $R_C \subseteq E_C$

.2 $D_C \subseteq E_C$ iff $R_C \subseteq S_C$

.3 Therefore, in all cases $D_C \cap R_C = \emptyset$

.4 *In the net* η *:* $R_C^{\bullet} \cap D_C = \emptyset$

Proof:

Suppose otherwise.

- $\exists x \in R_C : \exists y \in D_C :$ xFy
- xF_Cy by the definition of subnet
- $x \in \mathrm{dom}(F_C)$
- $x \in D_C$ by D4.1
- $x \in D_C \cap R_C$ by initial assumption
- But:
- $D_C \cap R_C = \emptyset$ by D3.1 and D4

<u>contradiction</u> □

.5 In the net η : $R_C \cap D^{\bullet}_C = \emptyset$, equivalent to .4

.6 $F_C \subseteq D_C \times R_C$, equivalent to .4

.7 In the net η : $D^{\bullet}_C = R_C$

Suppose otherwise, then:

- $\exists x \in D_C$: $\exists y \notin R_C$: xFy
- $\exists C' \in \mathcal{C}$: $x, y \in X_{C'}$ by D3.3
- $D_{C'} \cap D_C \neq \emptyset$ since they both contain x
- $D_{C'} = D_C$ by D4.1 and D3.2
- Then $R_{C'} = R_C$ by D4.3
- But $y \in R_{C'}$ <u>and</u> $y \notin R_C$

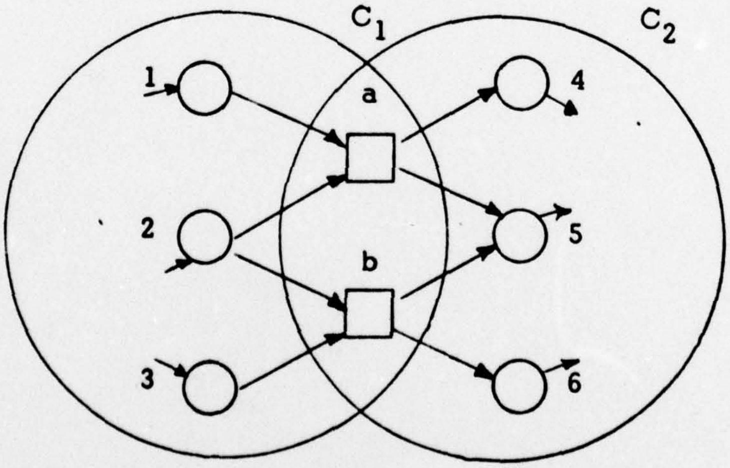
contradiction □

.8 In the net η : $D_C = {}^{\bullet}R_C$

proof analogous to .7

D6 A first example

.1



(In looking at this "parse" of the net into choices, remember D1.)

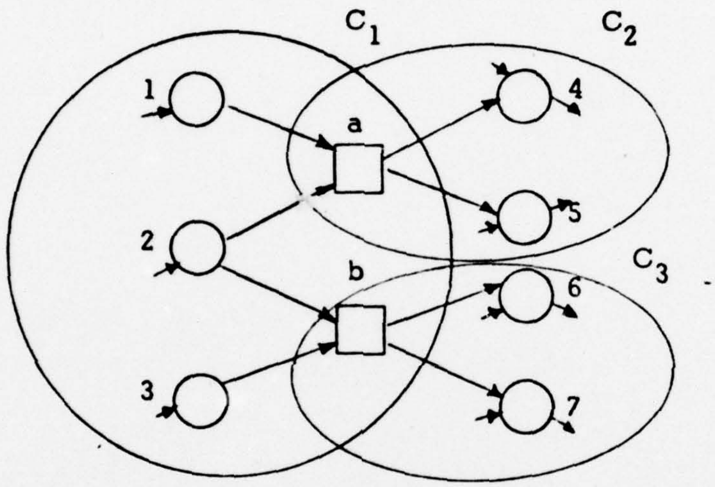
.2 $D_{C_1} = \{1, 2, 3\}$; $R_{C_1} = \{a, b\}$

$D_{C_2} = \{a, b\}$; $R_{C_2} = \{4, 5, 6\}$

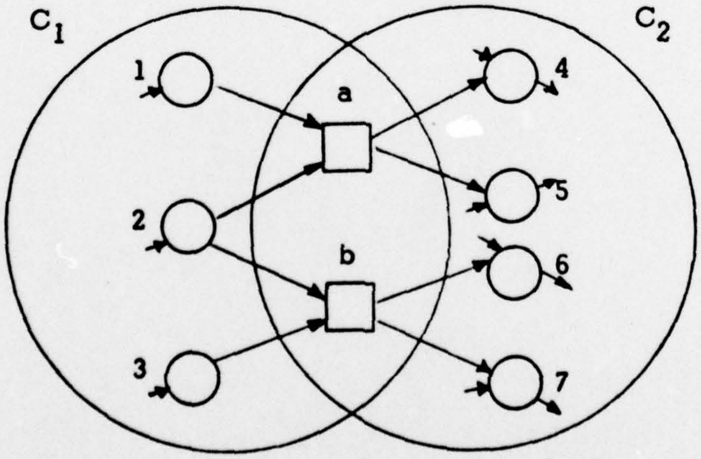
$C = \{C_1, C_2\}$ is the only choice structure which the net .1 supports. The next example can support either one of two choice structures.

D7

.1



.2



Comment: The "dual" of net .1 (interchanging boxes and circles) supports the analogous two choice structures.

Example D7 suggests what is true in general: that a net does not uniquely determine a choice structure (if, indeed, it supports any). But example D7 also suggests how alternative parsings into choice-subnets relate to each other: there is always a parse into smallest possible choice subnets. All other parses consist in grouping the smallest possible subnets into larger units. That this is all the variation in choice structuring that is possible for a given net is clear on the basis of the image offered in D1.

D8 Event-choices and state-choices

Our definitions and examples make clear that in a Petri-net two choice-types will be found.

- Those, whose R-set consists of event elements (and D-set consists of state elements). These are called event-choices.

- Those whose R-set consists of state-elements (and D-set consists of event elements). These are called state-choices.

Determinations of local state govern what local change ensues (event choice); determination of local change governs what local state ensues (state choice).

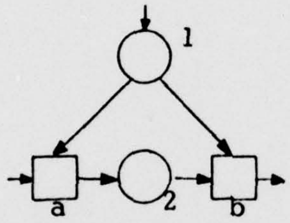
The example B2 had the form of an event choice (with its resultants going right and going left). The example read backwards has the form of a state choice. In its backward form the road fork is viewed as a road merge, and the signal does not steer the traveler, but records the direction from which he came.

D9 Net elements as choice mediators

Consider a net in which every element lies in both dom(F) and range(F). Suppose further that a choice structure for the net is given. Then D3.2 and D3.3 guarantee that every net element is part of exactly two distinct choices: in one of them it is a resultant, and thus helps to express the result of choosing; in the other one it is a determinant, and thus helps to express the determinants of choosing. In that sense, every net element acts as mediator between two particular choices.

D10 An example of a net which does not support any choice structure

.1

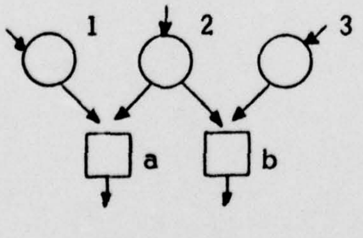


Element 1 feeds a and b. Thus a and b must both be resultants of the same choice. Element 2 feeds b; thus elements 1 and 2 must both be determinants of that same choice. Thus all four elements must belong to the same choice. But element a also feeds element 2 and thus element a must not only be a resultant, but also a determinant of the choice, in violation of D3.1.

D11 A comment on the relation between standard net interpretation and choice:

Consider the choice structure

.1



In line with the interpretations discussed under C3-C7, the figure .1 means that a and b compete for the input 2. If on a certain occasion a occurs, b is on that occasion excluded from occurring, and vice versa. Thus, according to the interpretations C3-C7 one might suppose that if markers "arrive" (by prior firings) at 1 and at 2 but not 3, then a determination exists which produces the result 'a occurs and b is excluded from occurring'. In the simulation game, such a "determination" may indeed come to exist - in the mind of the simulator - but the determination is not represented in the marking. The marking does not tell us that a marker might not yet arrive at 3 before the commitment to a is made, thus throwing the question of what is determined open.

D12 Final remark about net structure and choice structure

As we have seen, not every net supports choice structures in conformity with axioms D3. Thus axioms D3 implicitly specify a particular class of Petri-nets (comparable to the requirement for state-machine decomposability). But this is only the beginning of restrictive force on Petri-net structure which is (and will be) generated by the choice interpretation. It is hoped and expected that the formal structural restrictions on Petri-nets generated by the interpretations pertaining to choice will help to isolate a class of net structures mathematically tractable as well as representationally powerful.

E Determinations and Results

E1 Event-choice results

In its interpreted form, the result of an event-choice is that some event occurs and that the alternative events, as defined by the choice, are excluded from occurring. We express such a result by asserting, for each element of the R-set of a choice, whether it occurs or is excluded from occurring. We now define for all event elements the sentence forms:

.1 $\overset{+}{o}(x) \overset{\Delta}{=}$ the event (element) x occurs
$\bar{o}(x) \overset{\Delta}{=}$ the event (element) x fails to occur ,

With these, we can express the result of an event-choice in the form of a conjunctive sentence:

$$\bigwedge_{x \in G} \overset{+}{o}(x) \wedge \bigwedge_{x \in H} \bar{o}(x)$$

where $G \cup H = R$ and $G \cap H = \emptyset$

Formally, we can define a result r as the ordered pair of sets $\langle G,H \rangle$ such that a sentence of the form .1 asserts the result. If $r = \langle G,H \rangle$ we define $\overset{+}{r} \overset{\Delta}{=} G$ and $\overset{-}{r} \overset{\Delta}{=} H$. It is obvious that, if the R-set of a choice has n elements, then, there exists 2^n possible results. We shall think of $\overset{+}{r}$ as the set of determinants which have been assigned '+' value, and $\overset{-}{r}$ as the set of determinants which have been assigned '-' value.

E2 State-choice Determinations

The results of event-choices yield determinations for state-choices. Combinations of event elements that occur and are excluded from occurring determine what state elements next hold and are excluded from holding. A state-choice determination is expressed in terms of its D-set by a sentence of the form:

$$\bigwedge_{x \in G} \overset{+}{o}(x) \wedge \bigwedge_{x \in H} \overset{-}{o}(x)$$

where $G \cup H = D$ and $G \cap H = \emptyset$

In other words,, such a determination is given when the elements of the D-set are partitioned into two blocks: those event elements which occur and those which are excluded from occurring. In general, the partial specifications contributed by the separate D-set elements of a state-choice to its next resolution come from the last results of several distinct event-choices, as will be discussed more carefully below.

As in E1, if the D-set of a choice has n elements then there exist 2^n possible specifications. Also, as before, relative to a determination d , we will define $\overset{+}{d}$ to be the set G and $\overset{-}{d}$ to be the set H , as per .1 . Here again, $\overset{+}{d}$ is the set of determinants with '+' value, and $\overset{-}{d}$ the set of determinants with '-' value.

E3 State-choice results

In its interpreted form, the result of a state-choice is that some state holds and the alternative states, as defined by the state-choice, are excluded from holding. We express what holds with the help of the R-set of the choice by saying, for each element of the set, whether it holds or is excluded from holding. We now define, for all state elements, the sentence forms:

.1 $\overset{+}{h}(x) \overset{\Delta}{=}$ the state (element) x holds
.2 $\bar{h}(x) \overset{\Delta}{=}$ the state (element) x fails to hold

With these, we can express the result of a state-choice with a conjunctive sentence of the form

$$\bigwedge_{x \in G} \overset{+}{h}(x) \wedge \bigwedge_{x \in H} \bar{h}(x)$$

where $G \cup H = R$ and $G \cap H = \emptyset$

Formally, the result r is taken to be the ordered pair $\langle G,H \rangle$ (just as in E1) with $\overset{+}{r} = G$ and $\bar{r} = H$.

E4 Event-choice specifications

The results of state-choices determine the results of subsequent event-choices. Combinations of states that hold and fail to hold determine what events next occur and fail to occur.

An event-choice specification is expressed in terms of its D-set by a sentence of the form

$$\bigwedge_{x \in G} \overset{+}{h}(x) \wedge \bigwedge_{x \in H} \bar{h}(x)$$

where $G \cup H = D$ and $G \cap H = \emptyset$

In general the partial specifications contributed by the separate elements of the D-set of the event-choice to its next resolution come from the last results of several distinct state-choices, as will be discussed below.

F. System Boundary

The meaning of a net, $\eta = \langle S, E, F \rangle$ as a description of a system Σ, critically depends upon assumptions as to how the environment of Σ can act upon it, and how it can act upon its environment. These assumptions can be expressed as properties assumed to hold for all nets η' representing systems Σ' which consist of Σ, extended by some part of some one of its possible environments. Such nets η' will contain η as sub-net, and can be thought of as "extensions" of η. Thus we can say: the meaning of a given net depends (among other things) on the class of extensions of it that the describer means to allow.

F1 Definitions

Given a net $\eta = \langle S, E, F \rangle$, $X = S \cup E$, and a set $G \subseteq X$, we now define:

.1 G is post-complete within η if, in all intended extensions η' of η, ($G^{\bullet}$ in η') = ($G^{\bullet}$ in η). Otherwise, G is called post-partial.

.2 G being pre-complete and pre-partial are defined in an exactly analogous manner.

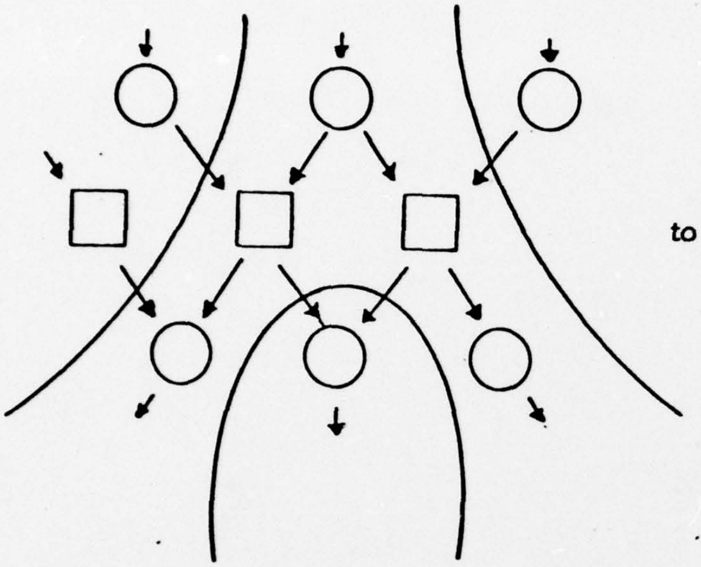
F2 If, for a net, it is known for each element $x \in X$ whether x is post-complete or not and pre-complete or not, then we naturally know the corresponding facts for every subset of net elements. In the sequel we shall assume, for all nets, that this information is known for each of its elements with the help of the following constructs and notations.

.1 We define for a net η, two subsets of its elements X :

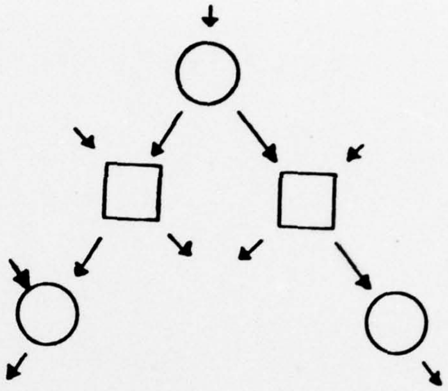
$$X^{\circ} = \{ x : x \text{ is post-complete in } \eta \}$$
$$^{\circ}X = \{ x : x \text{ is pre-complete in } \eta \}$$

.2 In the graphic form of η : if a net element x is post-partial, we attach to the graphic symbol - circle or square - a short arrow pointing out of the symbol; if it is post-complete we do not attach such a short arrow; similarly, if x is pre-partial we attach to the graphic symbol a short arrow pointing into the symbol; if it is pre-complete we do not. This notation has the advantage that it arises naturally, if one thinks of a given net as resulting from an excision from a larger net.

F3 **Example:**



to yield:



G. The Compatibility Relations

Prior to each choice resolution there is a process of accumulating determinant values (from prior and adjacent choice resolutions) until a determination of the next result is present. Once it is present, it can be re-written as that next result.

If the choice lies at the system boundary, and some of its determinants lie in the environment then, even after all of the determinants within the net have been assigned '+' or '-' value, the next result will only be partially specified. In a system simulation game, the simulator would then be free to choose any of the several results compatible with the determination as far as it is known to represent the outcome of choosing.

Conversely, if some of the choice resultants lie in the environment, then the choice result, as far as it is represented within the system will only partially specify the determination which produced it.

If, on the other hand, neither determinants nor resultants lie in the environment, then each determination (if coherent) specifies a result uniquely, and each result (if coherent) specifies a determination uniquely (i.e. the one which produced it). In short, determination and result are informationally equivalent.

The compatibility relation as defined below deals with choices whether they are on the boundary or not. It thus must take into account whether the determinants are post-complete and whether the resultants are pre-complete. In fact, the axioms for compatibility help to give definition to what it means to be on the boundary.

We shall first exhibit the conditions for compatibility in a form most favorable to inspecting its component meanings. Subsequently, we will re-express it in condensed and calculationally useful forms. We shall render the meanings in the language of "input/output" as introduced in C3.

G1 *Definition of compatibility*

Let d and r be a determination and a result of a choice C with D-set D and R-set R. If C is an event choice then:

$$d \ \underline{\text{comp}} \ r \overset{\Delta}{=} (\forall x \in D)\ (\forall y \in R):$$

.1 $y \in \overset{+}{r} \Rightarrow {}^{\bullet}y \subseteq \overset{+}{d}$

If (in the result) *y occurs, then* (in the determination) *all inputs of y must be present.*

.2 $y \in \overset{-}{r} \cap {}^{\circ}X \Rightarrow {}^{\bullet}y \not\subseteq \overset{+}{d}$

If (in the result) *y fails to occur, then* (in the determination) *some input of y must fail to be present.*

We can only be guaranteed to see such a missing input if y is pre-complete.

.3 $x \in \overset{+}{d} \cap X^{\circ} \Rightarrow x^{\bullet} \not\subseteq \overset{-}{r}$

Every presence (in the determination) *must contribute to some occurrence* (in the result).

We can only be guaranteed to see such an occurrence if the state-element that is present is post-complete.

Comment: .3 guarantees that, if x is post-complete, changing its value from '+' to '-' has an effect on the result.

If C is a state choice then:

$$d \ \underline{\text{comp}} \ r \overset{\Delta}{=} (\forall x \in D)\ (\forall y \in R):$$

.4 $x \in \overset{+}{d} \Rightarrow x^{\bullet} \subseteq \overset{+}{r}$

If (in the determination) x occurs, then (in the result) all outputs of x must be present.

.5 $x \in \bar{d} \cap X^{\circ} \Rightarrow x^{\bullet} \not\subseteq \overset{+}{r}$

If (in the determination) x fails to occur, then (in the result) some output of x must fail to be present.

We can only be guaranteed to see such a missing output if x is post-complete.

Comment: .5 guarantees that, if x is post-complete, then changing its value from '-' to '+' has an effect on the result.

.6 $y \in \overset{+}{r} \cap {}^{\circ}X \Rightarrow {}^{\bullet}y \not\subseteq \bar{d}$

Every presence (in the result) must have some occurrence as its source (in the determination).

We can only be guaranteed to see such an occurrence if the state-element that is present is pre-complete.

G2 Immediate consequences of G1

.1 G1.1 is equivalent to:

$$x \in \bar{d} \Rightarrow x^{\bullet} \subseteq \bar{r}$$

.2 G1.4 is equivalent to:

$$y \in \bar{r} \Rightarrow {}^{\bullet}y \subseteq \bar{d}$$

Thus, if C is an event choice, the conditions for d comp r specify when $^{\bullet}y \subseteq \overset{+}{d}$, when $^{\bullet}y \not\subseteq \overset{+}{d}$, when $x^{\bullet} \subseteq \bar{r}$ and when $x^{\bullet} \not\subseteq \bar{r}$. Similarly if C is a state choice, the conditions for d comp r specify when $x^{\bullet} \subseteq \overset{+}{r}$, when $x^{\bullet} \not\subseteq \overset{+}{r}$, when $^{\bullet}y \subseteq \bar{d}$ and when $^{\bullet}y \not\subseteq \bar{d}$.

G3 The compatibility relation and net simulation

With Petri-net interpretations as per C3-C8, system states are represented by state markings. A single marker type is used to designate the state elements that hold as part of the system state, while unmarked states are assumed not to hold. Transformations of system state are represented by event firings which transport markers from event inputs to event outputs. To play the "choice resolution" game on a net, various changes in these conventions are involved.

.1 "System states" - more aptly called "time slices" - must now be represented by a distribution over the net elements of two types of markers: plus-markers and minus-markers. Furthermore, the markers are no longer restricted to reside on state-elements only, but may appear on either kind of net element. Thus a marking tri-partitions the net elements.

.a ⊕, ⊞ : now determined to hold (occur)

.b ⊖, ⊟ : now determined to fail to hold (occur)

.c ○, □ : not now determined
[One may also say: not a part of the determination of what now holds (occurs) or fails to hold (occur).]

A simple example will make vivid the distinction between .b and .c. A work station normally goes through a fixed cycle of activity: fetch; process; deliver. Suppose the station fetches, processes, and delivers binary values. When it fetches 0, fetch 0 is marked plus and fetch 1 is marked minus; but deliver 0 and deliver 1 are unmarked, because they do not serve to define what occurs (or doesn't) at fetch time.

.2 The unit of "firing" is not an event element, but a choice. When all the determinants of a choice are marked with some distribution of plus and minus, we have a choice determination which, if coherent, can be replaced by a compatible choice result. If the choice is not at the system boundary, there will only be one compatible choice result; otherwise several, and the simulator must supply some additional choice determinants so as to produce a choice resolution.

G4 Condensed statement of the compatibility relation

If C is an event choice:

$$d \ \underline{\mathrm{comp}} \ r \overset{\Delta}{=} \begin{cases} .1 & \bar{r} \supset \bar{d}^{\bullet} \supset \bar{r} \cap {}^{\circ}X \\ .2 & \overset{+}{d} \supset {}^{\bullet}\overset{+}{r} \supset \overset{+}{d} \cap X^{\circ} \end{cases}$$

.1 is equivalent to G1.2 and G2.1
.2 is equivalent to G1.1 and G1.3

If C is a state choice:

$$d \ \underline{\mathrm{comp}} \ r \overset{\Delta}{=} \begin{cases} .3 & \overset{+}{r} \supset \overset{+}{d}^{\bullet} \supset \overset{+}{r} \cap {}^{\circ}X \\ .4 & \bar{d} \supset {}^{\bullet}\bar{r} \supset \bar{d} \cap X^{\circ} \end{cases}$$

.4 is equivalent to G1.4 and G1.6
.5 is equivalent to G1.5 and G2.2

G5 The compatibility relation and "mutual exclusion"

The compatibility relation as defined in G1 does not express the idea that if n events are in competition over a common input at most one of them can occur. That requirement for a "compatible" result could be expressed in the following form.

For event choices:

.1 $x \in \overset{+}{d} \Rightarrow \exists$ at most one element y: $y \in x^{\bullet} \cap \overset{+}{r}$

It would also imply a re-wording of the interpretation under G1.3, as follows:

Every presence (in the determination) must contribute to <u>exactly one</u> occurrence (in the result).

Similarly, the force of the "safety" assumption discussed in C8.1 is not embodied in the compatibility relation. It could be added in the form:

For state choices:

.2 $y \in \overset{+}{r} \Rightarrow \exists$ at most one element x: $x \in {}^{\bullet}y \cap \overset{+}{d}$

Also, the interpretation under G1.6 would be reworded as follows:

> Every presence (in the result) must have exactly one occurrence as its source (in the determination).

Both of these missing restrictions on compatibility have something to do with "mutual exclusion": mutual exclusion of events which compete for an input in the case of event choices, and mutual exclusion of events which compete for the production of an output in the case of state choices.

In their end-effect, our construction will embody both of these mutual exclusion ideas, but in the form of coherence properties of determinations and results. In the examples of the compatibility relation which will shortly follow, the reader will find determinations and results declared as compatible which criteria .1 and .2 would throw out.

G6 Finding results compatible with a given determination

The definition G4 is not in a convenient form for finding results compatible with a given determination (if any exist). On the basis of G4 it is easy to prove the correctness of the following criteria, adapted to this purpose.

If C is a choice with R-set R then:

If C is an event-choice, the result r is compatible with the determination d iff:

.1 $(R - \bar{d}^{\bullet}) \supseteq \overset{+}{r} \supseteq (R - \bar{d}^{\bullet}) \cap {}^{\circ}X$

.2 ${}^{\bullet}\overset{+}{r} \supseteq \overset{+}{d} \cap X^{\circ}$

If C is a state-choice, the result r is compatible with the determination d iff:

.3 $(R - \overset{+}{d}^{\bullet}) \supseteq \bar{r} \supseteq (R - \overset{+}{d}^{\bullet}) \cap {}^{\circ}X$

.4 ${}^{\bullet}\bar{r} \supseteq \bar{d} \cap X^{\circ}$

G7 Examples of finding results from determinations

There follow our first set of examples of finding the results which are compatible with a given determination. The procedure for finding them will be based on G6.1-.2. This procedure may be considered as a format for formal "causal reasoning" - from determinations of state to the occurrence of events. The principal purpose of these examples is to show the subtlety and inherent complexity in this reasoning - in particular, to show how critical to this reasoning is what the reasoner knows about the post-completeness of determinants and the pre-completeness of resultants.

Given a determination of an event-choice, what results are compatible? A natural procedure based on G6.1-.2 is the following.

Step 1 finding $(R - \bar{d}^{\bullet})$ - i.e. finding all event elements which are not excluded from occurring by the given determination

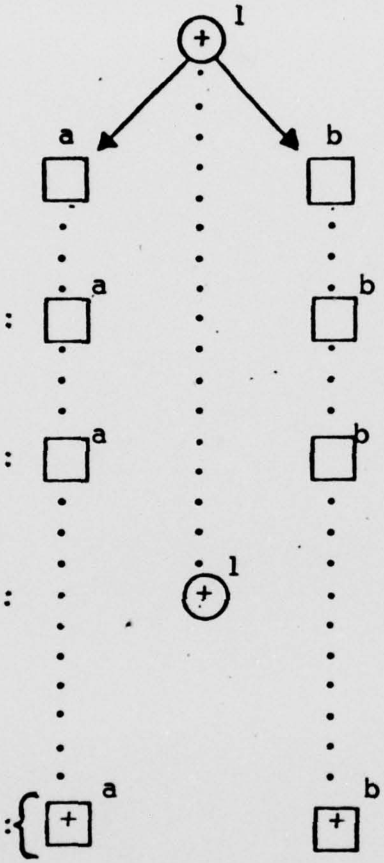
Step 2 finding ${}^{\circ}X \cap (R - \bar{d}^{\bullet})$ - i.e. finding all event elements which cannot be excluded from occurring, no matter what the environment determines

Step 3 finding $\overset{+}{d} \cap X^{\circ}$ - i.e. finding the set of all state-element holdings which must result in some occurrence that is part of the described result.

The first two steps "bracket" $\overset{+}{r}$ between "can occur" and "must occur", while the last step expresses an additional constraint which $\overset{+}{r}$ must satisfy.

G8 Examples

.1



A choice with a determination

Step 1 $(R - \bar{d}^{\bullet})$: event-elements which are not excluded from occurring by the given determination

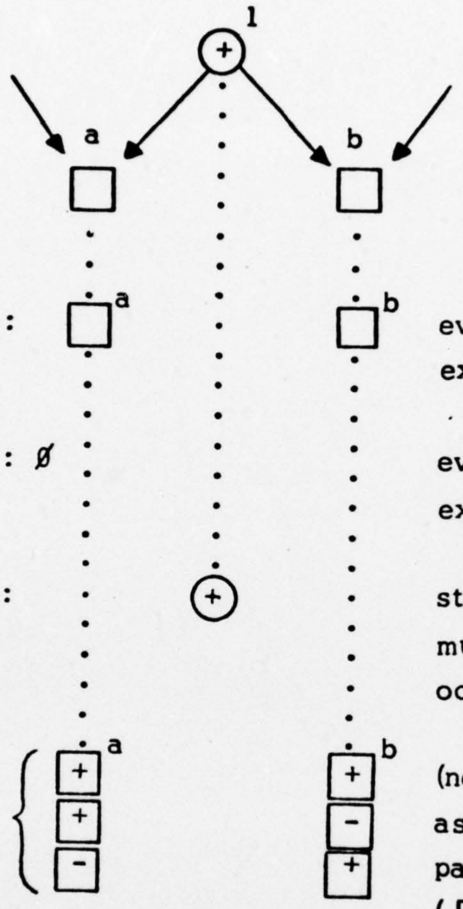
Step 2 $^{\circ}X \cap (R - \bar{d}^{\bullet})$: event-elements which cannot be excluded from occurring no matter what the environment determines

Step 3 $\overset{+}{d} \cap X^{\circ}$: *state*-element holdings which must result in an event-element occurrence that is part of the described result.

possible results :

Note: this result is, of course, incompatible with mutual exclusion. But since a and b are pre-complete, there is nothing imaginable which could prevent either of them from occurring once state-element 1 holds.

.2



A choice with determination

Step 1 $(R - \bar{d}^{\bullet})$: event-elements which are not excluded

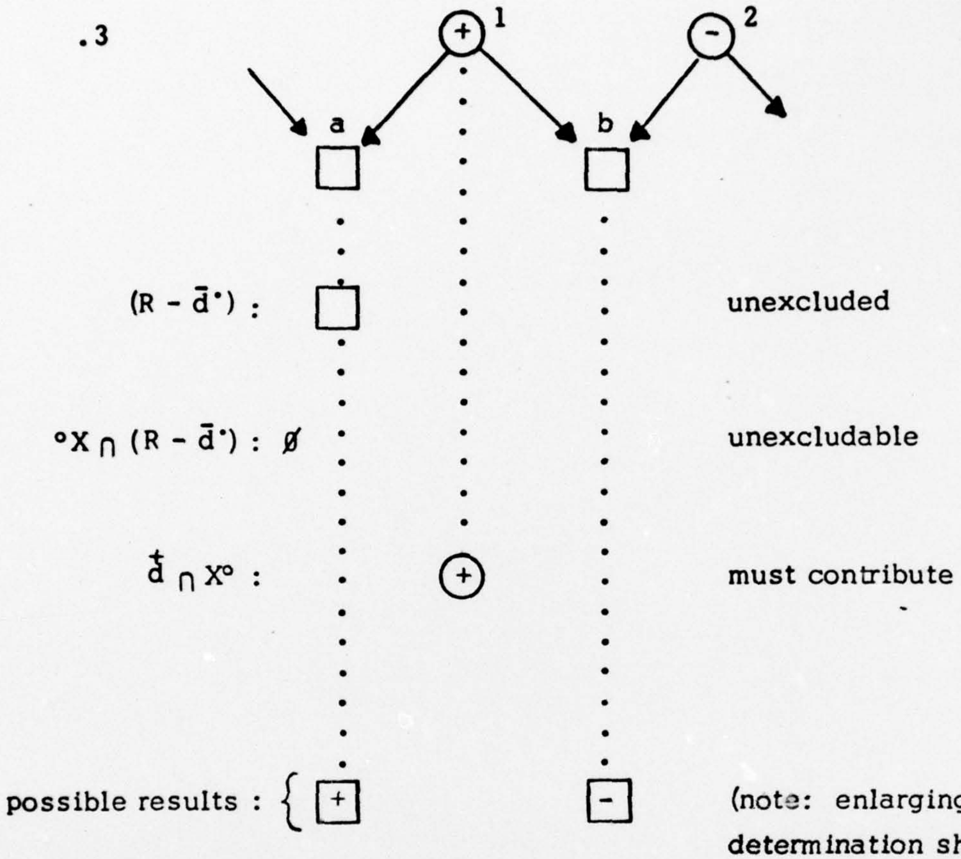
Step 2 $^{\circ}X \cap (R - \bar{d}^{\bullet}) : \emptyset$ event-elements which cannot be excluded

Step 3 $\overset{+}{d} \cap X^{\circ}$: state-elements holdings which must result in an event-element occurrence

possible results

(note: modifying this example by assuming that state element 1 is post-partial would mean adding the result ($\boxed{-}\ \boxed{-}$) to the set of possible results.)

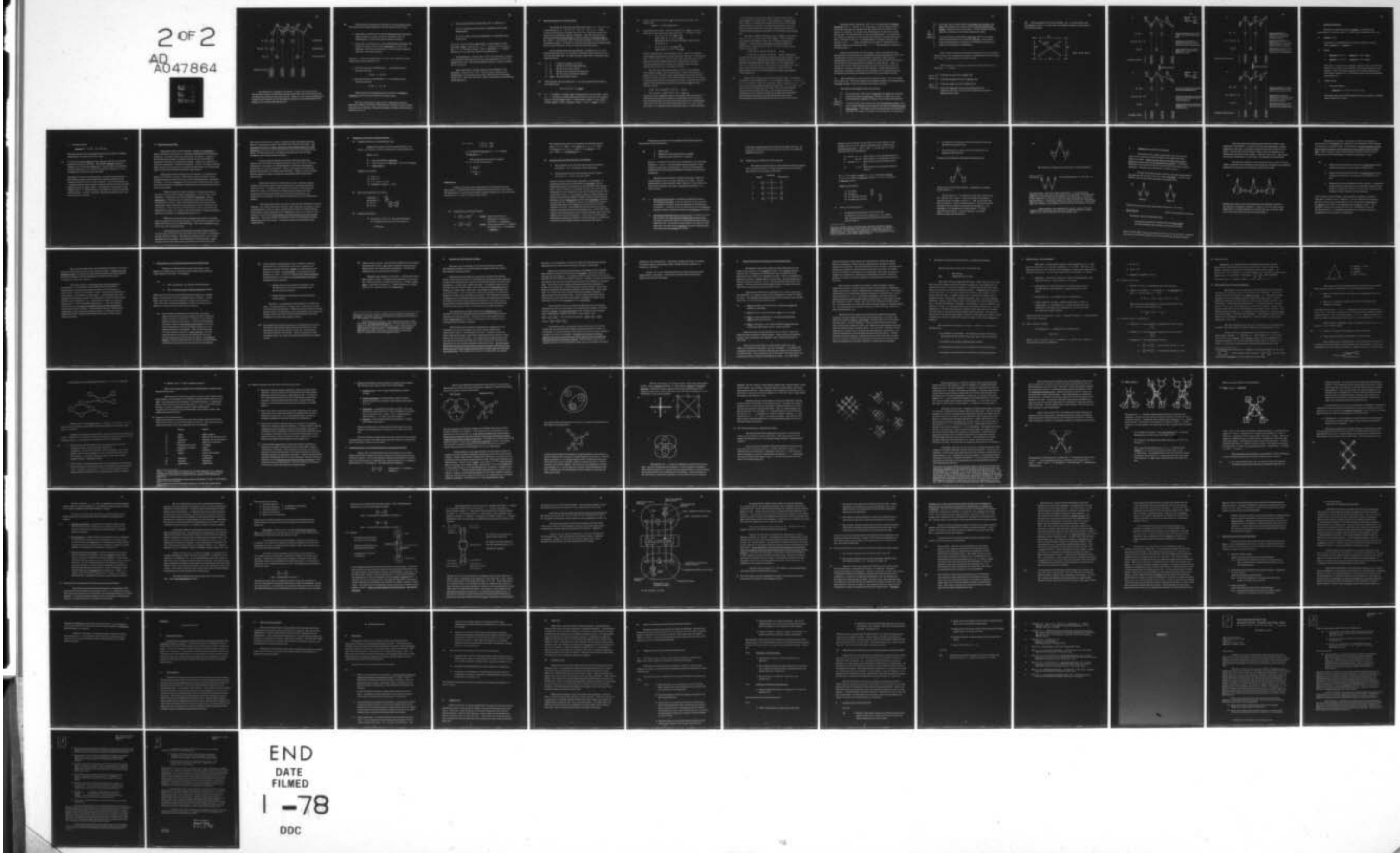
.3


1
2
a
b
$(R - \bar{d}^{\bullet})$:
unexcluded
$^{\circ}X \cap (R - \bar{d}^{\bullet}) : \emptyset$
unexcludable
$\overset{+}{d} \cap X^{\circ}$:
must contribute
possible results :
+
-

(note: enlarging the partial determination shown in this example so as to inhibit the uninhibited (but inhibitable) event would yield a specification which is compatible with no result at all.)

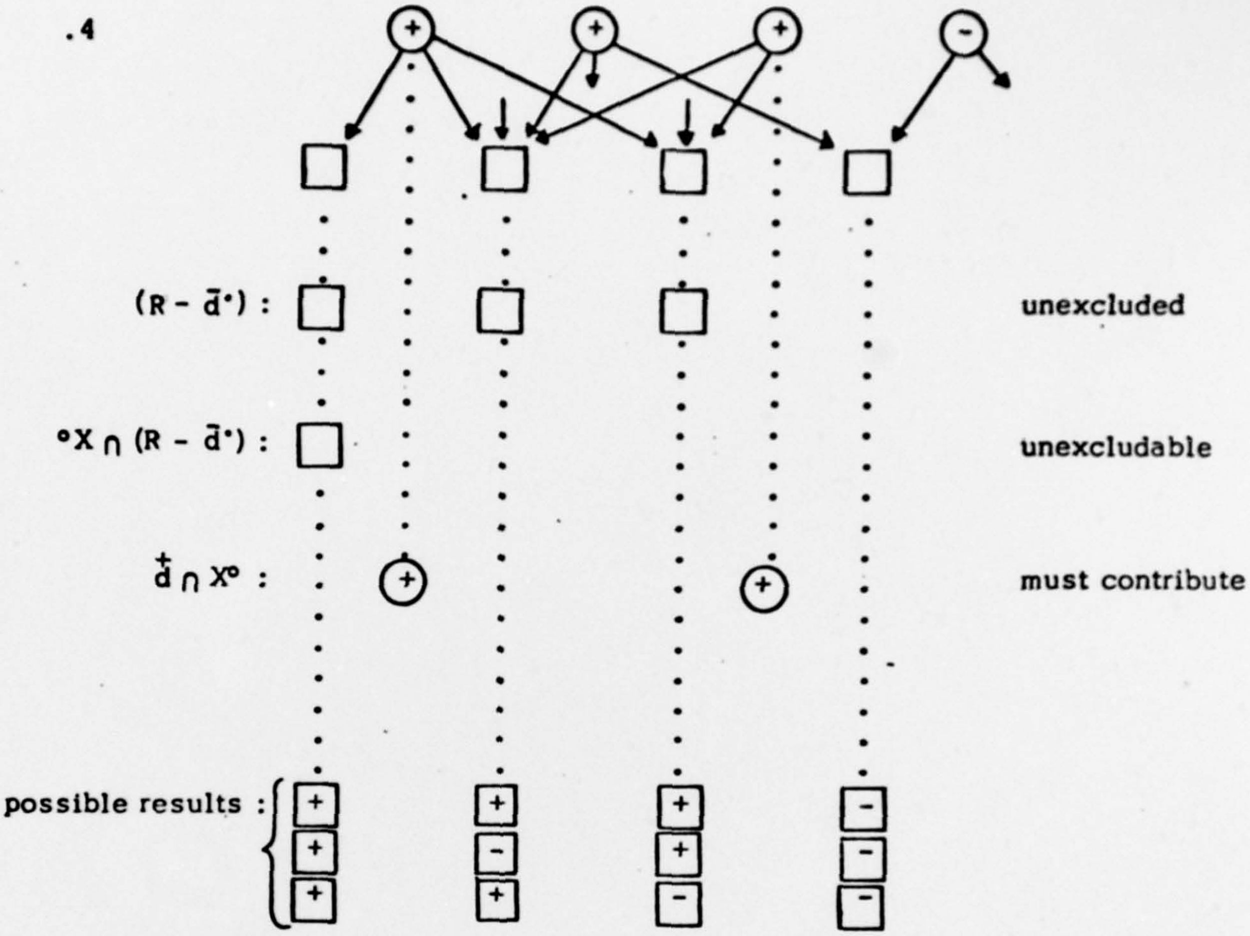
AD-A047 864 MASSACHUSETTS COMPUTER ASSOCIATES INC WAKEFIELD F/G 9/4
RESEARCH ON INFORMATION SYSTEM SPECIFICATION.(U)
AUG 77 A W HOLT N00014-76-C-0781
UNCLASSIFIED CADD-7708-0911 NL

2 OF 2
AD
A047864



END
DATE
FILMED
1 -78
DDC

.4


$(R - \bar{d}^{\bullet})$:
unexcluded
$^{\circ}X \cap (R - \bar{d}^{\bullet})$:
unexcludable
$\overset{+}{d} \cap X^{\circ}$:
must contribute
possible results :
+
+
+
+
−
+
+
+
−
−
−
−

Our example set is biased in two senses: it deals with event-choices, and not state-choices; it exemplifies finding results from given determinations and not finding determinations from given results. Thanks to a set of net transformations (called sense reversals) we will find that our examples can be mechanically transformed so as to remove these biases (H5).

G9 Two questions concerning compatibilities between determinations and results are of special importance, from the applied point of view:

.1 under what circumstances is a given determination not compatible with any result, or a given result not compatible with any determination? (a part of the system liveness question)

.2 under what circumstances is a given determination compatible with exactly one result, and thus fully determines the result; and similarly, under what circumstances is a given result compatible with exactly one determination?

Question .1, going from determination to result, has a definitive answer which follows at once from G6.1-.4.

.3 For event-choices: a determination d is compatible with at least one result iff:

$$\overset{+}{d} \cap X^{\circ} \subseteq {}^{\bullet}(R - \bar{d}^{\bullet})$$

.4 For state-choices: a determination d is compatible with at least one result iff:

$$\bar{d} \cap X^{\circ} \subseteq {}^{\bullet}(R - \overset{+}{d}^{\bullet})$$

These properties of determinations are examples of coherence properties which will be the subject of a section below.

As to the circumstances under which a determination uniquely determines a result, or a result a determination, we shall, for now, content ourselves with a sufficient, though not necessary condition which follows at once from G4.1-.4.

.5 For any choice (event or state) with D-set D and R-set R :

If $R \subseteq {}^{\circ}X$ then at most one result is compatible with a given determination.

If $D \subseteq X^{\circ}$ then at most one determination is compatible with a given result.

(If $R \subseteq {}^{\circ}X$ and, for a given determination d there exists a result r such that $d \underline{\text{comp}}\ r$ then; by G4.1, $\bar{d}^{\bullet} = \bar{r}$ for event-choices and, by G4.3, $\overset{+}{d}{}^{\bullet} = \overset{+}{r}$ for state-choices. Thus d determines r . A similar argument works in going from results to determinations.)

By interpretation, a choice with $R \subseteq {}^{\circ}X$ is one for which, relative to its results as described by the R-set, the determination is fully described by the D-set. The full determination should, of course, uniquely specify the result.

Less self-evident, but also implied by our definitions is the following: a choice with $D \subseteq X^{\circ}$ is one for which, relative to its determination, as described by the D-set, the full result is described by the R-set. Then, this full result uniquely determines the determination which produces it.

H. Net Transformations: Sense Reversals

The axioms for Petri-nets (see C1) assure that if $\eta = \langle S, E, F \rangle$ is a net, then so are $\eta' = \langle S, E, F^{-1} \rangle$ and $\eta'' = \langle E, S, F \rangle$. Each of these transformations if applied twice in a row, yields the original net, and so may be reasonably called a "reversal". We will call the first of these transformations time reversal tr and the second one element reversal, er. Given η , we will write tr(η) for the time reversal of η and er(η) for its element-reversal. We shall now consider what effect these transformations have on choice structure, as far as it has been described above.

Let us consider what sets and relations, in addition to S, E, and F must be associated with a net in order to encompass choice structure as far as it has been described. We shall list them in the order in which they were introduced above.

H1	.1	C	a family of subnets, as per D3
	.2	$\mathcal{D}$	the set of all choice determinations
	.3	$\mathcal{R}$	the set of all choice results
	.4	X°	the set of post-complete net elements
	.5	$^{\circ}X$	the set of pre-complete net elements
	.6	comp	the compatibility relation

H2 Taken together with the original net η we can represent the structure by the ordered tuple:

$$\langle \eta, C, \mathcal{D}, \mathcal{R}, X^{\circ}, {}^{\circ}X, \underline{\mathrm{comp}} \rangle$$

H3 We further define a "reversal" neg of determinations and of results, which maps determinations to determinations and results to results: by reversing the positive and negative parts. In other words, taking d to be $\langle \overset{+}{d}, \bar{d} \rangle$, we have neg(d) = $\langle \bar{d}, \overset{+}{d} \rangle$; similarly, taking $r = \langle \overset{+}{r}, \bar{r} \rangle$ neg(r) = $\langle \bar{r}, \overset{+}{r} \rangle$.

H4 Lastly, we define the relation $\underline{\overline{\text{comp}}}$ between determinations and results thus:

$$d \ \underline{\overline{\text{comp}}} \ r \equiv \text{neg}(d) \ \underline{\text{comp}} \ \text{neg}(r)$$

H5 The question now arises: Given $\langle \mathcal{N}, \mathcal{C}, \mathcal{D}, \mathcal{R}, X^{\circ}, {}^{\circ}X, \underline{\text{comp}} \rangle$, can one, in some sense, apply $\underline{\text{tr}}$ and $\underline{\text{er}}$ to the entire tuple? The answer is yes.

1. $\langle \mathcal{N}, \mathcal{C}, \mathcal{D}, \mathcal{R}, X^{\circ}, {}^{\circ}X, \underline{\text{comp}} \rangle \xrightarrow{\underline{\text{tr}}}$
   $\langle \underline{\text{tr}}(\mathcal{N}), \underline{\text{tr}}(\mathcal{C}), \mathcal{R}, \mathcal{D}, {}^{\circ}X, X^{\circ}, \underline{\text{comp}} \rangle$
   ($\underline{\text{tr}}(\mathcal{C})$ names the set of nets obtained by applying $\underline{\text{tr}}$ to each net in $\mathcal{C}$)
2. $\langle \mathcal{N}, \mathcal{C}, \mathcal{D}, \mathcal{R}, X^{\circ}, {}^{\circ}X, \underline{\text{comp}} \rangle \xrightarrow{\underline{\text{er}}}$
   $\langle \underline{\text{er}}(\mathcal{N}), \underline{\text{er}}(\mathcal{C}), \mathcal{D}, \mathcal{R}, X^{\circ}, {}^{\circ}X, \underline{\overline{\text{comp}}} \rangle$

We are asserting that, if the first tuple in H5.1 and H5.2 represents a net together with all choice-related structure, then so does the second tuple in H5.1 and H5.2. The only part of this assertion which is not obvious in that the transformation of the compatability relation is correctly expressed in H5.1 and H5.2. But this is easy to verify with the help of G4.1-.4, as we shall now show.

First note that both reversals leave $\mathcal{D} \times \mathcal{R} \cup \mathcal{R} \times \mathcal{D}$ unchanged. Now suppose d $\underline{\text{comp}}$ r, for an event-choice. Under time-reversal, as defined in H5.1, ${}^{\circ}X$ and X° become interchanged; every determination becomes a result and every result a determination; every pre-set becomes a post-set and every post-set becomes a pre-set. Applying these substitutions to the expressions which define the relation $\underline{\text{comp}}$ for event choices, namely

$\bar{r} \supseteq \bar{d}^{\bullet} \supseteq \bar{r} \cap {}^{\circ}X$ and $\overset{+}{d} \supseteq {}^{\bullet}r^{+} \supseteq \overset{+}{d} \cap X^{\circ}$, we get

$\bar{d} \supseteq {}^{\bullet}\bar{r} \supseteq \bar{d} \cap X^{\circ}$ and $\overset{+}{r} \supseteq \overset{+}{d}^{\bullet} \supseteq \overset{+}{r} \cap {}^{\circ}X$, which we observe to be the conditions which a specification and result pair of a state-choice must satisfy in order to be compatible with one another. But time-reversal also converts every event-choice into a corresponding state-choice and thus the original compatible pair $\langle d, r \rangle$, now viewed as a pair $\langle r', d' \rangle$ relative to the transformed choice, is still compatible.

In exactly analogous manner one gets the same result by starting with d comp r for a state-choice in the original net. Finally since time-reversal is involutary (the reverse of the reverse is the identy transformation) we see that, if a determination/result pair is compatible in the transformed net, then the same pair (now viewed as a result/determination pair) must also be a compatible in the original net.

Now let us consider element-reversal. Suppose we are given a pair ⟨d, r⟩ for an event choice with d comp r. What conditions are therefore satisfied by the pair ⟨d', r'⟩ where d'=neg(d) and r=neg(r') ? We find these by transforming the expressions for compatibility by interchanging $\overset{+}{d}$ and $\bar{d}$, as well as interchanging $\overset{+}{r}$ and $\bar{r}$. Thus, from

$\bar{r} \supseteq \bar{d}^{\bullet} \supseteq \bar{r} \cap {}^{\circ}X$ and $\overset{+}{d} \supseteq {}^{\bullet}\overset{+}{r} \supseteq \overset{+}{d} \cap X^{\circ}$ we get

$\overset{+}{r} \supseteq \overset{+}{d}^{\bullet} \supseteq \overset{+}{r} \cap {}^{\circ}X$ and $\bar{d} \supseteq {}^{\bullet}\bar{r} \supseteq \bar{d} \cap X^{\circ}$ which, once again,

we observe to be the compatibility conditions for state-choices. Thus, the pair ⟨d', r'⟩ is a compatible pair for the transformed choice. An analogous argument yields the same result, if we had begun with ⟨d,r⟩ as a *compatible pair of a state-choice in the original* net. Once again, since element-reversal is involutary, we also have the compatibility of ⟨d, r⟩ in the transformed net implies compatibility in the original net.

H6 What is the meaning of time-reversal transformation. Two natural possibilities present themselves. If η describes a system Σ , then tr (η) describes a system Σ' that runs backwards. This "running backwards" should not be confused with putting a motor in reverse gear. If Σ is an engine that diminishes its fuel supply as it runs, then Σ' is an engine that increases its fuel supply as it runs. This is thermodynamically unsound and we may expect the ability to express that unsoundness by finding the restrictions on choice-related structures which have appropriate real interpretations and will not admit time reversal.

The other natural meaning for tr(η) is, as an expression for thinking backwards about Σ. The net η is, in any case, a tool for thinking about Σ, and we may associate this thinking with the pursuit of causal chains in the manner of a predictor, who reasons from determinations of choices to results. The net tr(η) may then be taken as a tool for thinking about Σ in the manner of a detective ("post-dictor") who reasons from results to the determinations of them. A simulation step in η is interpretable as a thought of the form 'and therefore, next ......' while a simulation step in tr(η) is interpretable as a thought of the form 'and therefore, last ......' -- both thoughts in reference to the same system Σ.

With this interpretation of time-reversal, what is the meaning of the fact that time reversal leaves the compatibility relation unchanged (as per H5.1)? It means that the rules for causal reasoning "backwards" (from results to determinations) about event-choices are the same as the rules for causal reasoning "forwards" (from determinations to results) for state-choices. (And, of course, the rules for reasoning backwards for state-choices are the same as the rules for reasoning forwards for event-choices.) Among other things this shows that, as regards a single choice (of whatever type it might be) the rules for reasoning backwards are different than the rules for reasoning forwards.

H7 What interpretations may be given to element reversal? We have nothing to say about this. But we can express the interpreted content of the transformation from comp to comp$^-$ which is associated with element reversal.

The rules for compatibility imply the following:

For Event-Choices

.1 If we know that a state-element does hold then some event-element in its (total) post-set must occur, although we don't know which - for that generally depends on holding or not of other pre-conditions.

.2 If we know that a state element has been excluded from holding, then we are certain that all event-elements in its post-set will be excluded from occurring. Thus negative local knowledge does not contribute to uncertainty of outcome, but rather to the resolution of uncertainties.

For State-Choices

.3 If we know that an event-element is excluded from occurring then some state-element in its (total) post-set must be excluded from holding, although we don't know which - for that generally depends upon the occurrence or not of other predecessor events.

.4 If we know that an event-element does occur then we are certain that all state-elements in its post-set will hold. Thus positive local knowledge does not contribute to uncertainty of outcome but rather to the resolution of uncertainties.

Thus, if state-choices are formally transformed into event-choices and vice-versa, the compatibility relation is transformed by the interchange of '+' and '-' in specifications as well as results.

The following is a suggestive example illustrating the ideas expressed in .1 - .4 above.

Event-Choice

.5 If the door is open then it might slam

.6 If the door is not open then it will not slam

State-Choice

.7 If the door does slam then it will be shut

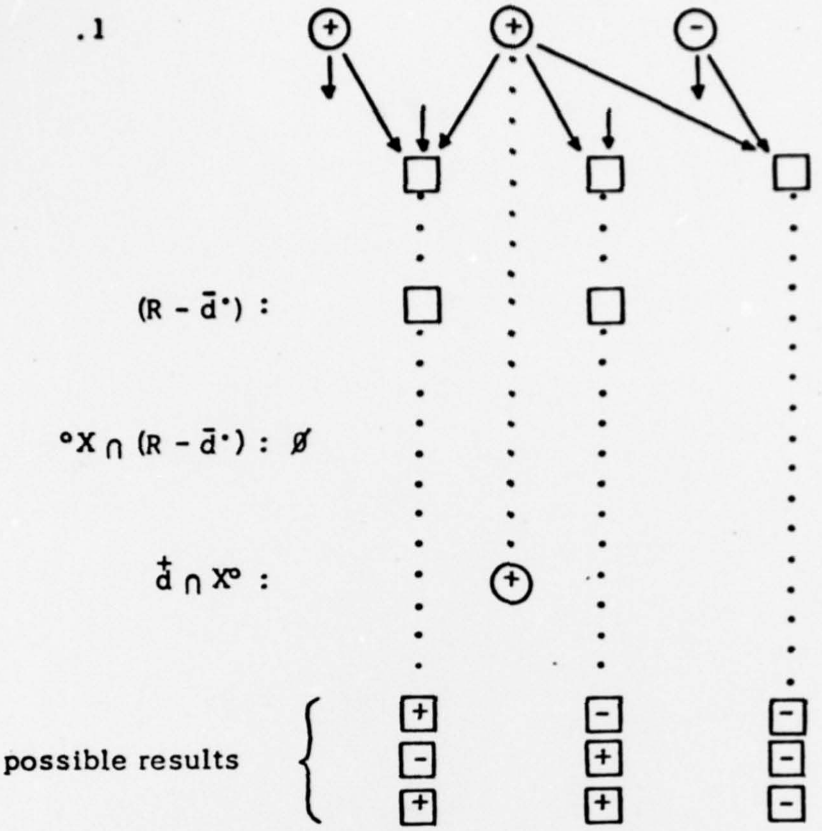
.8 If the door does not slam then it might not be shut
(additional example: If they hadn't pumped his stomach he might not have lived)

H8 We now present a set of four examples, H8.1-.4 in the style of G8. The examples of H8.1-.4 are the time- and element-reversals of each other in the following pattern:

$$\begin{tikzcd}
\text{H8.1} \arrow[r, leftrightarrow, "\underline{tr}"] \arrow[d, leftrightarrow, "\underline{er}"'] \arrow[dr, leftrightarrow, "\underline{er}\cdot\underline{et}"] & \text{H8.3} \arrow[d, leftrightarrow, "\underline{er}"] \arrow[dl, leftrightarrow, "\underline{er}\cdot\underline{et}"'] \\
\text{H8.2} \arrow[r, leftrightarrow, "\underline{tr}"] & \text{H8.4}
\end{tikzcd}$$

Note: $\underline{er}\cdot\underline{et} = \underline{et}\cdot\underline{er}$

.1



↔ time reversal

H8.3 on next page

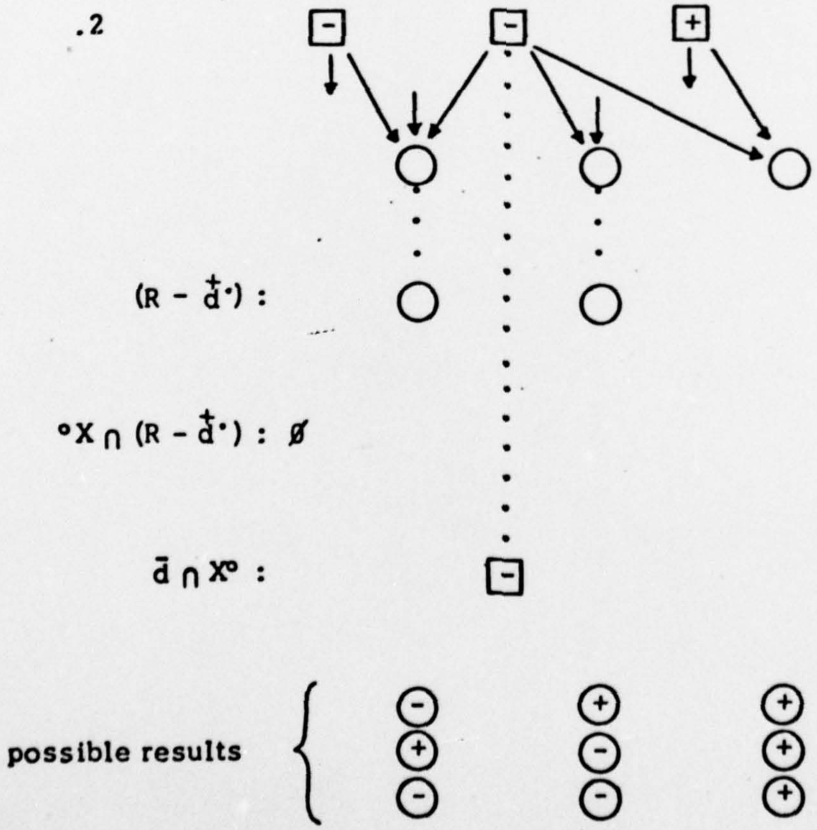
(R − d̄·) : <u>are not excluded</u> from occurring by the given determination

°X ∩ (R − d̄·) : ∅ — <u>cannot be excluded</u> from occurring, no matter what the environment determines

$\overset{+}{d}$ ∩ X° : <u>must result in some occurrence</u> that is part of the described result

element reversal

.2



↔ time reversal

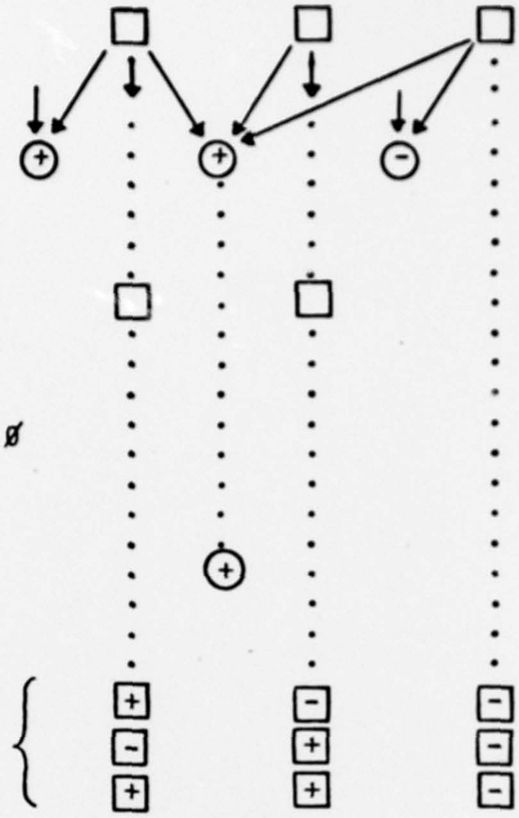
H8.4 on next page

(R − $\overset{+}{d}$·) : <u>are not included as holding</u> by the given determination

°X ∩ (R − $\overset{+}{d}$·) : ∅ — <u>cannot be included as holding</u>, no matter what the environment determines

d̄ ∩ X° : <u>must result in some state exclusion</u> that is part of the described result

.3



$(D - \cdot\bar{r})$: are not excluded as having occurred by the evidence presented by the given result

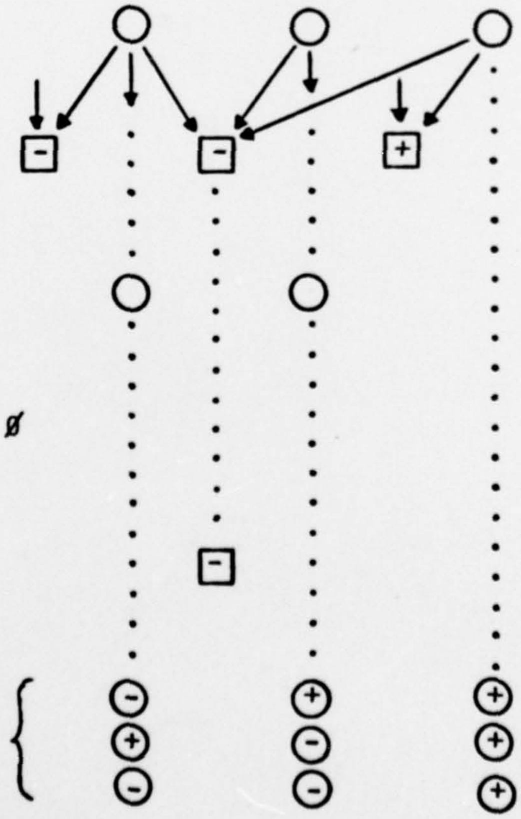
$(D - \cdot\bar{r}) \cap X^\circ : \emptyset$ cannot be excluded as having occurred, no matter what additional evidence the environment presents.

$^\circ X \cap \overset{+}{r}$: must be the result of some occurrence that is part of the described determination

possible determinations

element reversal

.4



$(D - \cdot\overset{+}{r})$: are not included as having held by the evidence presented by the given result

$(D - \cdot\overset{+}{r}) \cap X^\circ : \emptyset$ cannot be included as having held, no matter what additional evidence the environment presents

$^\circ X \cap \bar{r}$: must be the result of some exclusion that is part of the described determination

possible determinations

J. Coherence Properties

We shall now define the property coherent, as applied to the determinations $\mathcal{D}$ and the results $\mathcal{R}$, of the choices associated with a net.

J1 $\underline{\text{coherent}} \subseteq \mathcal{D} \cup \mathcal{R}$

To assert that a determination or a result is coherent, we shall write $\underline{\text{coherent}}(d)$ or $\underline{\text{coherent}}(r)$.

J2 Axioms

.1 $\underline{\text{coherent}}(d) \equiv \exists r \in \mathcal{R} : \quad \underline{\text{coherent}}(r) \text{ and } d\ \underline{\text{comp}}\ r$

.2 $\underline{\text{coherent}}(r) \equiv \exists d \in \mathcal{D} : \quad \underline{\text{coherent}}(d) \text{ and } d\ \underline{\text{comp}}\ r$

Thus every d which is not compatible with any result is not coherent, and any result which is not compatible with any determination is not coherent. Thus we know that coherent event-choice determinations must satisfy condition G9.3, and coherent state-choice determinations must satisfy G9.4.

J3 Further axioms

.1 For event-choices:

$$\underline{\text{coherent}}(r) \Rightarrow \forall y \in \overset{+}{r} : ({}^{\bullet}y)^{\bullet} \cap \overset{+}{r} = \{y\}$$

This means that no two event-elements that occur as part of a coherent result compete for an input.

.2 For state-choices

$$\underline{\text{coherent}}(d) \Rightarrow \forall x \in \overset{+}{d}: \ {}^{\bullet}(x^{\bullet}) \cap \overset{+}{d} = \{x\}$$

This means that no two event elements that occur as part of a coherent determination both produce the same output.

J4 If we add the property coherent to the structure tuple H2, we can still apply time-reversal to the tuple. The property coherent remains unchanged, just as the relation comp does. We can see at once that this works by noting that if we make all the notational changes in J3.1 for reversing time we get J3.2, and vice versa.

The same is not true of element reversal! No natural transformation of the coherence property will represent the coherence property for the element reversed net with its choice structure. Thus in the presence of a coherence demand strong enough to yield the effects of mutual exclusion, element reversal no longer "works". This failure is made visible in the examples H8.1-.4. Note that for H8.1 the first two results are coherent, but not the third. For H8.2, however, (the element-reverse of H8.1) all three results are coherent.

IV. Concurrency and Choice

This paper focuses on two concepts - choice and concurrency - which I believe are necessary to a scientific understanding of information processes. The paper addresses several kinds of interest - interest in (a) the theoretical aspect of computer science; (b) the construction of simulation languages; (c) technical philosophy, insofar as it concerns itself with "information"; (d) Petri-net research. Finally, the ideas presented offer a window into an area of work which has been active for at least eight years as part of the Information Systems Theory Project.

Concurrency means something like co-presence - co-presence of objects or activities or states - while choice means something like the potential presence of some one out of a set of possible objects, activities, or states. A choice is resolved in that one of the possibles becomes actual. As will develop below, concurrency and choice have something significant to do with one another.

These concepts find their reflection in propositional logic in the form of the connectives and and exclusive or - but it is only a pale reflection. Concurrency and choice interrelate phenomena, and not propositions. Phenomena are, by definition bound to time and place, as propositions are not. That is why logical connectives can be explicated with no direct reference to time and space; not so with choice and concurrency. That is also why an explication of choice and concurrency necessarily involves, so I believe, the consideration of temporal sequence, which has no reflection in propositional logic at all.

While we are well primed to recognize the relevance of choice and its resolution to information concepts, we are largely unprepared to see what concurrency has to do with information. Still in an introductory vein, let me say a few words about this.

The consequence of communication between several actors is the coordination of their actions and states with one another - coordination in time, place and content. So long as two actors are two - i.e. at all separable from one another - this coordination will not be total - some freedom of action and state of the one relative to the other must exist.

That relative freedom may be great, if they are widely separated from each other in "communication space" and small if they are near each other. The concurrency relation is a formal expression of the relative freedom of state and action owing to the spatial separation of distinct but intercommunicating actors. Put yet another way: concurrency expresses the lack of those constraints which communication produces.

It is our ultimate objective to express the above ideas (and whatever goes with them) in mathematical forms which will make an effective contribution to the design and implementation of information transforming, transmitting and storing mechanisms. Below, some steps in this direction will be demonstrated, and further steps described. Finally, it will be argued that the expected utilities are worth the considerable effort which these steps require.

About eight years ago I chose Petri-nets as a starting symbolic vehicle with which to explore system concepts and to develop formalized versions of them. In regard to this aspect of my activity, the drawing of Petri-net pictures, and imagining or executing token games over these pictures has played the same role as the drawing of geometrical constructs has for a geometer, or the writing of equations has for a mathematical analyst.

From the strictly formal view point, Petri nets are mathematical objects definable by a small collection of axioms. They admit quite a variety of styles of interpretation as well as formal superstructures useful to systems thinking. In this paper I shall concentrate on an interpretation which is, I believe, best adapted to understanding the mechanics of communication at its most elementary level. This is the setting in which the issue of the relationship between concurrency and choice arises. We shall now present our definitions and interpretations.

A. Preparatory Definitions and Interpretations

A1 Standard Definition of Directed Petri Nets

Except for the names of the net-elements that I use below, the following is Petri's definition of a directed Petri net.

$$\eta \triangleq \langle R, A, F \rangle$$

R : a set of elementary resources

A : a set of elementary activities - aslo called actions

F : a flow relation

Subject to the axioms

.1 $R \cup A \neq 0$

.2 $R \cap A = 0$

.3 $F \subseteq R \times A \cup A \times R$

.4 $\text{domain}(F) \cup \text{range}(F) = R \cup A$

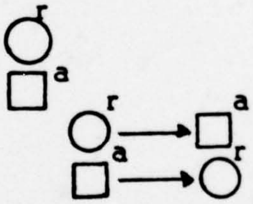
A2 Petri nets graphically represented

a resource r :

an activity a :

$\langle r, a \rangle \in F$:

$\langle a, r \rangle \in F$:



A3 Auxiliary definitions

.1 For a net $\eta = \langle R, A, F \rangle$ we reserve the letter X to designate the set of all net elements:

$$X \triangleq R \cup A$$

.2 $\forall x \in X: \quad x^{\bullet} \triangleq \{y : xFy\}$

$\qquad\qquad\qquad\;\; {}^{\bullet}x \triangleq \{y : yFx\}$

$x^{\bullet}$ is called the *post-set* of x ; ${}^{\bullet}x$ is called the *pre-set* of x .

We can extend this concept to subsets $Y \subseteq X$, in the usual manner:

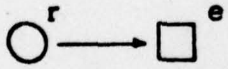
$$Y^{\bullet} \triangleq \bigcup_{x \in Y} x^{\bullet}$$

$${}^{\bullet}Y \triangleq \bigcup_{x \in Y} {}^{\bullet}x$$

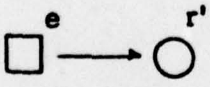
Interpretation

I shall now present some interpretations of net elements and the flow relation. I believe that these interpretations taken as basis will support the development of a rich class of system concepts adequate to a very wide range of applications , as shall be briefly argued below.

A4 Interpretation of the Flow Relation

.1 

means: every occurrence of e consumes an instance of r - put briefly: e *consumes* r

.2 

means: every occurrence of e produces an instance of r' - put briefly: e *produces* r'

This interpretation leads us to introduce the following natural terms: If $r \in {}^{\bullet}e$ then r is an input of e; if $r \in e^{\bullet}$ then r is an output of e; if $e \in {}^{\bullet}r$ then e is a source of r; if $e \in r^{\bullet}$ then e is a destination of r.

A5 An axiom about co-presence and co-occurrence

.1 Two instances of one-and-the-same resource cannot be co-present (i.e. be present at the same time).

.2 Two occurrences of one-and-the-same activity cannot co-occur (i.e. occur at the same time).

At first, A4 and A5 sound highly restrictive. For example: in computing environments memory cells are resources, yet not ordinarily thought of as consumed by the activities which require them. What is more two "co-present" memory calls are certainly candidates for "two instances of one-and-the-same resource" thus running afoul of A5.1. This strongly suggests system contexts in which it may be natural enough to think in terms of resources and activities, but not elementary resources and elementary activities as per A4 and A5. I believe, however, that under a wide class of circumstances there exist useful levels of descriptions which satisfies A4 and A5. Some examples directly below are aimed at making it plausible that such descriptions exist. At the end of this paper we shall briefly discuss what applied benefits might come from finding such levels of description. In any case, the descriptive restrictions A4 and A5 are useful to us in this paper as a stage-set for examining the relation between concurrency and choice.

Elementary resources are to be distinguished semantically from one another in three dimensions:

A6 .1 What is it?
.2 Where (in system space) does it appear?
.3 When (in system time) does it appear?

However, in the Petri-net model we are discussing, two elementary resources, r and r' , can only differ from one another in their activity sources and/ or destinations. Thus all the dimensions of difference A6 must reflect themselves in choices of source and destination.

We are quite accustomed to relate the question 'what is it' to the questions 'how is it made' and 'how is it used'. We are unaccustomed to think similarly about 'where is it' and 'when is it'; the next two examples show that, although unaccustomed, it is quite possible.

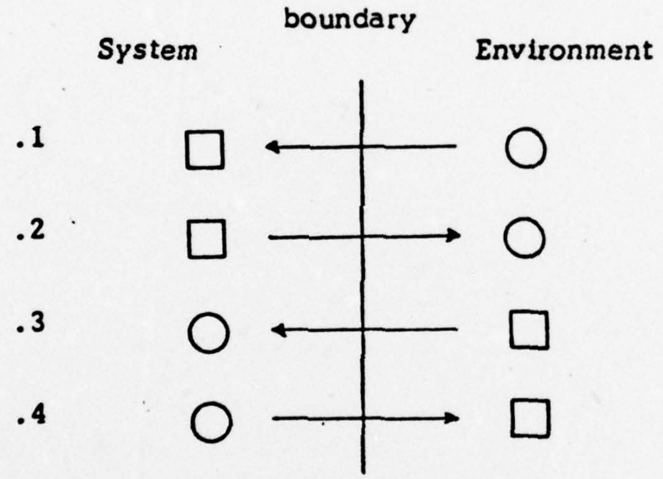
A7 .1 File A in the file drawer is a different elementary resource than File A on the desk. Both are instances of the same file, but enablers of different activities (because of the difference in location). A transfer of the file from drawer to desk consumes the first of these elementary resources and produces the second.

.2 The closed drawbridge when no car is on it is a different elementary resource than the closed drawbridge when a car is on it. The former enables the activity of drawbridge opening while the latter does not. The activity which brings a car onto the bridge when none was there before consumes the first of these elementary resources and produces the second.

A few more interpretations and axioms are in order, but first, we must add something to Petri-net structure which is not ordinarily regarded as part of it.

A8 Connection of a System to its Environment

We shall not assume that our nets represent isolated systems. For our present purposes, we shall assume that the environment can connect to the system in four ways:



Formally we can represent the boundary of a net by adding, to the ordered triple $\langle R, A, F \rangle$ another structural given: namely a pair of distinguished sets I and E with $I \subseteq R \cup A$ and $E \subseteq R \cup A$ with the following interpretation

.5 activity $e \in I \equiv$ there exists an environmental input of e

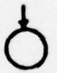
.6 $e \in E \equiv$ there exists an environmental output of e

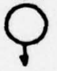
.7 resource $r \in I \equiv$ there exists an environmental source of r

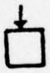
.8 $r \in E \equiv$ there exists an environmental destination of r

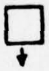
if $r \in I$ it is called an *import;* if $r \in E$ it is called an *export;* if $e \in I$ it is called an *importing* activity; if $e \in E$ it is called a *exporting* activity.

Graphic representation

.9 an import: 

.10 an export: 

.11 an importing activity: 

.12 an exporting activity: 

A7 Axioms and Interpretations

.1 If (instances of) all inputs of an activity are co-present then the activity *must* occur. Put another way: all that can prevent an activity from occurring is missing inputs.[1]

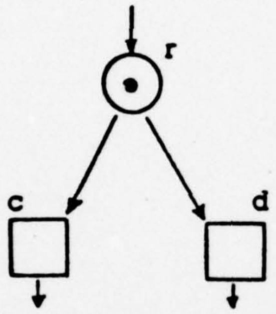
[1] To those already familiar with Petri-nets viewed as structures whose behaviors can be simulated by the well-known "firing rule", it should be clear that A7.1 deviates from the commonly accepted interpretation of event occurrence. The common interpretation is: if all inputs of an event are co-present then the activity *can* take place.

.2 An instance of a resource can only be produced once and can only be consumed once.

.3 Two instances of a resource must be separated by an absence of that resource.

A7 leads to the following consideration (among others).

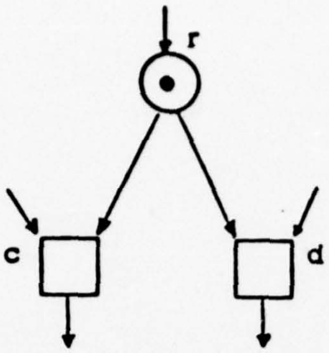
A8



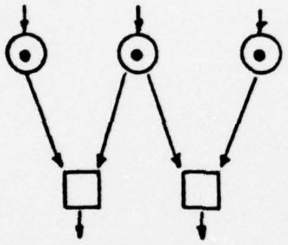
The black dot in the circle labelled r represents an instance of the resource r .

All the inputs of c are present (in this case, the instance of r must be 'all', since c does not import) and therefore c must take place; the same is true of d . But, if they both take place, resource r must be consumed twice! Thus the situation pictured in A8 must either be declared as meaningless, or as meaningful, but representing a system failure. We shall adopt the latter course. By way of contrast, no system failure is implied by the picture:

A9



(but neither is a failure of the same type as in A8 excluded!) [1]



[1] The picture is our representation of, what has, in the recent years, come to be called "the glitch" - a situation which commonly arises when one of two actions is to take place, depending upon which of two concurrently arriving signals "gets there first". (In the time of the scholastics, a situation of this kind was described and discussed by Jean Buridan. He posited a hungry man placed between two "identically attractive dishes", each exactly as easy to reach as the other. Under these ideal circumstances, Jean Buridan said that the man would starve to death.)

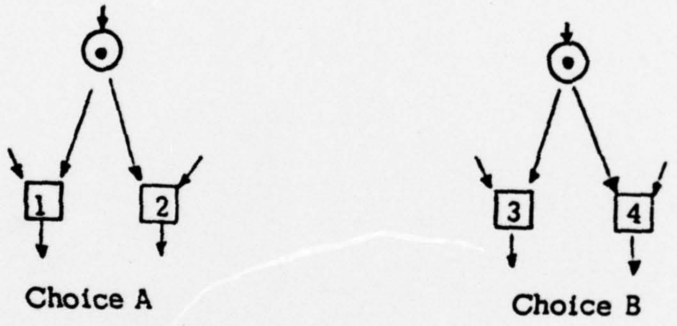
More generally, the potential for a "glitch" exists whenever a discrete state discrimination is required to determine whether a certain action is or is not to take place by some definite time.

B Concurrency as it Relates to Choice

It is often true in system contexts that one activity takes place to the exclusion of other activities which might have taken place in its stead. Under this circumstance the taking place of an activity is the expression of the resolution of a choice - a resolution governed by what inputs are and are not available at the time of choice resolution. We shall call such choices 'activity choices'.

If a pair of activity choices, A and B, can ever be concurrently resolved then the choices A and B are called concurrent. A pair of such choices ready for concurrent resolution is pictured below.

B1



And now the key point in this paper about concurrency and choice.

The CC Axiom*

If choices A and B are concurrent then

No possible resolution of choice A has any direct effect on what resolutions are possible for B (and vice versa).

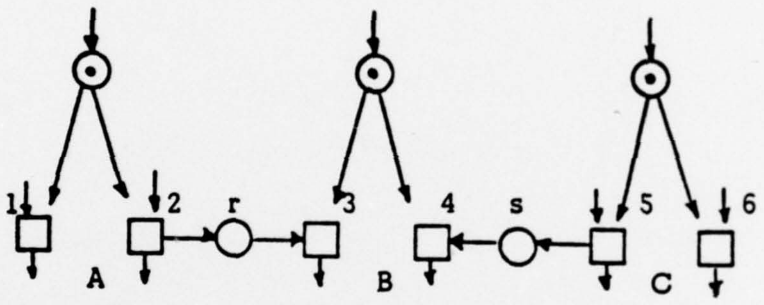
Such a "direct effect" exists if an activity which can be chosen under A produces or consumes an input (or output) of an activity which can be chosen under B.

*'CC' for Concurrency and Choice

This input/output disjointness gives structural content to the expression "direct effect", as also to the following topologically oriented expression of the idea: two choices cannot be concurrent if they are "too close" to each other (see the general remarks about concurrency in the introductory paragraphs of this paper). If they are "too close", they cannot be resolved independently of each other, and therefore not concurrently.

We shall now examine in some detail the case of pairs of choices which, according to the CC axiom do not satisfy the conditions for concurrency because the resolution of one produces an input to the other.

B2



Although this picture with our interpretation only explicitly introduces two sequence constraints (2 before 3; 5 before 4), the CC axiom denies concurrency as possible between any activity that can be chosen under A or C with any activity that can be chosen under B.

The effect is to be this: how choices A and C are resolved on this occasion is to determine how choice B is to be resolved on this occasion if a resolution is possible at all. In other words: that the outcome of choice B is to depend causally on the outcomes of choices A and C and on nothing else. (Only if activity 3 and 4 were importing activities would there still be structural room for dependence on something additional.) Thus:

B3

.1 Choice B will be resolved in favor of 3 if and only if choice A is resolved in favor of 2 and choice C in favor of 6.

.2 Choice B will be resolved in favor of 4 if and only if choice A is resolved in favor of 1 and choice C is resolved in favor of 5.

.3 Choice B will not be resolved at all (the "glitch", as explained in the footnote to A9) if choice A is resolved in favor of 2 and choice C is resolved in favor of 5 (it will also not be resolved if choice A and/or choice C are not resolved).

In the absence of the CC axiom - and therefore no sequence constraints other than "2 before 3" and "5 before 4" - we must weaken the connection between cause and result described in B3 by (a) replacing 'if and only if' by 'only if' in B3.1 and .2; (b) replacing 'if' by 'only if' in B3.3. (For example: Choice A is resolved in favor of 2, while choice C has not yet been resolved; r is produced and choice B is resolved in favor of 3, independently of how choice C is resolved.)

Thus an extra causal factor, not represented explicitly in choices A and C and their connections to B, has entered the scene - namely the factor of relative timing: how soon (if at all) does the effect of the next resolution of choice C influence the next resolution of choice B relative to the corresponding effect from choice A.

This "extra factor" is both technically and philosophically unacceptable. It is technically unacceptable because it destroys the possibility of tracing the outcomes of choices to the outcomes of other choices - forwards or backwards in time - one of the key objectives, in my opinion, of an adequate system model. It is philosophically unacceptable for the following reason: the extra factor of relative timing comes from the lack of constraint - *i.e.* *concurrency* as between the possible production of r and the possible production of s . But, as explained at the beginning, concurrency expresses the lack of those constraints which *communication produces*. Communication and only communication establishes causal connections between choices. Concurrency was to express the relative freedoms that remain in the light of these relative causal constraints.

C Characteristics of a Theoretical Framework for the CC Axiom

Although our discussion has built on Petri-nets, it has depended on expressions which have no self evident formal definition in the context of Petri-nets. For example:

C1

.1 How, structurally, are activity choices defined?

.2 How are the occasions of choice resolution defined?

These questions and many other related ones are treated in a working paper of mine entitled "A Petri-net Based Theory of Choice". In this paper, new formal structures pertaining to choice are constructed "over" Petri-nets, and the net apparatus is given a new basic interpretation as summarized in the next series of points.

C2 In the standard interpretation of Petri-nets, the taking place of an event is the atomic change in terms of which all system changes are expressed. It consists in the replacement of its inputs by its outputs. In the new interpretation the atomic change is the resolution of a choice; it consists in the replacement of the pre-conditions which govern the choice outcome by that outcome. What is more two types of atomic change are posited: activity choices, resolving what activities are and are not next present on the basis of what resources are and are not now present, and resource choices, resolving what resources are and are not next present, on the basis of what activities are and are not now present.

C3 In the standard interpretation of nets an element of state as represented by a circle element in the net is thought of as having two possible statuses: holding, as represented by the presence of a token, and not holding, as represented by the absence of a token. In the new interpretation resource absence (corresponding to 'not holding') is subdivided into two distinguishable kinds of absence:

.1 absence because the question of whether it will be produced in time for its next potential uses has not yet been resolved.

.2 absence because the question has been resolved in the negative.

The case .1 is represented by the absence of a token; the case .2 is represented by a negative token, which does not exist in the standard interpretation. (When a resource choice is resolved, all those which are determined to be next available are marked with positive tokens; all those which are determined not to be next available are marked with negative tokens.)

C4 The resolutions of activity choices are expressed by the removal of positive and negative tokens from circle elements of a net, and placing them on box elements of the net - the resolution of resource choices, the reverse. Thus activities get marked as taking place or not, just as resources get marked as available, or not.

C5 Grossly stated, we can, with the help of negative tokens express the difference between "now absent, though it might now have been present" and "could not now be present", and that with reference both to resources and activities.[1]

Example with reference to activities - a processing station goes through a cycle: it receives a bit; processes it; puts out a bit. When it receives 0 it does not receive a 1, though, at this time, it might have received a 1. When it receives a 0 it does not put out a 1 because this is not output time.

[1] It is now time to admit that A7.1 above must be viewed as inaccurate, or incomplete, as already implied by the discussion of A8 and the footnote to A9. More exactly, A7 should say

> If all inputs of an activity e are co-present then the activity must occur, if anything definite occurs. We only know that e must occur if all activities which compete with e for one-or-more of its inputs are prevented from occurring at this time, as expressed by the presence of negative tokens. In the context of the standard Petri-net "firing rule" there is no locally defined circumstance when it is known that an activity must take place.

D. Summary and Highlights by Example

We began with an argument as to why the concurrency relation has something essential to do with models of communication and, hence, with models of information flow.

Consistent with the idea that concurrency expresses the freedom which communication constrains, we related the concurrence of two activities to their causal independence, and not to the temporal relations between their occurrences, as many other writers have done. If I regularly send you mail, you may very well read the letter I sent you yesterday concurrently with my writing you a letter today. (Nota bene: the content of my letter today must in no way depend on your reaction to my letter of yesterday, if concurrency between my writing and your reading is to exist.) That asserted concurrency is independent of the temporal relations between your reading and my writing: if a common clock were used to make the comparison, we might discover that you were finished reading before I had begun to write, or the reverse, or that for some period of time my writing and your reading overlapped.

Our constructions implied that the causal dependence of two activities could not be explicated without seeing activity occurrences as resolutions of activity choices. But if that is so, than neither can the causal independence (and, hence, concurrency) of two activities be explicated without reference to choice.

The CC axiom and associated interpretations suggested that the concurrency of activities depends upon their spatial separation in communication space. The converse is not true. Consider a long race track in the form of a ring, with various observation points along its length where a car may be seen to pass. In a mode of track operation where one and only one car races around the ring, there will never be concurrent car observations at the spatially separated observation points. But the spatial extension of the track and the spatial separation of these observation points does imply that modes of track operation exist in which there are concurrent car observations. This implies in turn that in a period when, on purpose, the track is to be devoted to a single car, effort must be spent on keeping additional cars for which there is room on the track, off the track.

What if two activities are too close to each other to be concurrent? They might be so close to each other as to coincide, and thus be parts of one-and-the-same activity. In customary thinking about activity coincidence, one does not worry about their spatial relations. Instead, one says of two activity occurrences that they are coincident just because they happened at the same time. This leaves out the question of whether they merely happened to happen at the same time, or whether they are perforce united. In any case, there is an inherent and damaging vagueness about the notion 'at the same time' as a relation between activities which are spatially separated - a point to which we will shortly return. As was already made clear above, we would say of two activities which merely "happened to happen at the same time" that they are concurrent and not coincident.

If two activities are not so close to each other so as to coincide, and yet not so far from one another as to possibly be concurrent, then they are either ordered or mutually exclusive to one another (in net-structural terms, depending on whether they relate to one another, thus □ → ○ → □ , or in one of the ways thus □ ← ○ --→ □ or thus □ → ○ ← □).

In describing above the possible temporal relations of my writing a letter and your (concurrently) reading a letter (one before the other, or overlapped), we tacitly accepted these temporal relations between spatially separate activities as meaningful. I believe that success with those endeavors for sake of which the work in this paper exists is incompatible with accepting this meaning as primitive. What meaning it may have in specific contexts must be constructed on the basis of an analysis of causal dependence and independence. This remark applies pari passu to the idea of a pair of activity occurrences which, though spatially separate, are temporally coincident.

Finally, the various relations between activities discussed above apply equally to resources. A discussion of them in that context goes beyond the bounds of this paper.

E. Results Expected From the Kind of Work Sampled Above

We expect to develop the ability to build system models over which we can exercise - not a logical calculus - but a causal calculus: that is to say, to recreate by calculation the flow of effect by causal necessity in the context of an information system. Here it is important to notice that systems - whether natural or man-made - can be thought of as a network of causal connection. (E.g., 'He was arrested because he trespassed'; 'The water runs into this container because I opened that valve'; 'The water is blue because the sky is blue', etc.).

Informational differences as between which causal connections are established in a system context may be differences in 'what' or 'where' or 'when' . Any of these three kinds may be causally related to any of these three kinds, as the following examples show.

D1 .1 What the address on the message is determines where the message is delivered

.2 When the relay closes determines what key is actuated

.3 When an item is deposited on a moving belt determines where on the belt it will be

.4 Where a dot appears in a roster determines what character is produced (the general idea of "place notation")

These phenomena are the ultimate basis for various "trade-offs" which are often practiced in system design - such as making an operation be more "time-consuming" but, therefore, less "space-consuming" or vice versa.

These interconvertibilities of informational differences that makes it so technically desirable - not say "necessary" - to operate with a modeling framework in which all informational differences are treated in a uniform manner - are an important part of the reason for accepting the unaccustomed modes of thought illustrated in A7 above. It is also these interconvertibilities which make it so unreasonable to divide the subject matter of information systems into the communication department and the computation department. From the informational point of view, a transformation of bits may implement an information transport from one "place" to another (as when a location is renamed) just as a transport of bits from one place to another may implement an information transformation (as when a shift of a number in a register is used to implement a multiplication of that number by the base).

The existence of models over which a "causal calculus" can be effectively exercised would dramatically improve our abilities in information system specification, design, implementation, modification, etc. The calculus would enable us to demonstrate important properties of specifications, design and implementations - e.g., consistency properties, proof that wished for dynamic relations between system parts are maintained, or that certain outputs (as specified by 'what', 'where', and 'when') only depend on certain inputs, etc. It would enable us to study the relation between design and implementation - *i.e., the maintenance* of the causal connection which the design specifies in every implementation of it.

We are still at the stage of constructing the conceptual/scientific framework for these hoped-for practical accomplishments - it is here, that the issues related to choice, concurrency, and their relations to one another find their place. We are, therefore, still far removed from having all the tools, manual and computer-assisted, for modeling and calculating which will make the practical goals a present reality. It is our claim, however, that the conceptual/ scientific work, of which we have exhibited a small part, is an absolute prerequisite for the practical abilities discussed above. Without this basis we shall continue, in the future as we have in the past, to build ever-more complex and extensive information systems with ever-less understood consequences to the human societies in which they are installed.

V. Net Models of Organizational Systems, in Theory and Practice

The two key items in the title of this paper are

Net Models
and Organizational Systems

The "nets" that are meant are Petri-nets. Petri-nets are a new symbolic tool in the general domain of applied mathematics, for expressing and manipulating certain classes of meanings - meanings that are of prime importance to the practice of known as system analysis. "Organizational systems", on the other hand, is an awkward phrase for something that doesn't, as yet, have a graceful name. It refers principally to "systems" as understood by system analysts, programmers, computer designers, etc., but with a special emphasis: namely, such systems seen in the context of a human organization - e.g., an organ of business, production, or government. In such contexts, the effect of a system is to establish orderly and dependable connections - causal connections - between the decisions, actions, inputs and outputs, of people who play roles in the organizational context being considered. We therefore wish to orient our system descriptions towards an understanding of these effects - to enhance our ability to relate what is in the machine with what is around the machine.

This paper does not present "a result". Rather, its purposes are the following:

- to introduce to the reader, with the help of small illustrations, the general manner in which nets carry their burden of meaning

- to develop the concept "organizational system"

- to demonstrate relevance of the foregoing to systems practice

- to discuss the achievements and hopes of "applied net theory".

A What are nets, and net-models?

Petri-nets, as their name suggests, were discovered by C.A. Petri, and first described in his doctoral dissertation [1] in 1962. The theory and practice associated with them has since undergone a considerable development via many contributors. We can characterize Petri-nets boradly, as follows

A1 .1 Abstractly, they may be viewed as a class of mathematical structures governed by a small collection of axioms

.2 Associated with these structures is a certain principal style for their graphic representation - though subsidiary styles exist.

.3 Associated also, is a general style of interpretation

.4 Finally there are, for nets, certain characteristic modes of manipulation - or transformation - whose interpretations vary according to the underlying interpretations given to the nets, in a particular applied context.

We will now briefly describe A1.1 and .2. Section B covers A1.3 while sections C and D give examples.

A2 Nets, Abstractly defined

A directed net η, is defined as an ordered triple

$$\langle S, E, F \rangle$$

where S and E are sets, and F a relation, $F \subseteq S \times E \cup E \times S$ subject to the following axiomatic restrictions

.1 $S \cup E \neq \emptyset$

.2 $S \cap E = \emptyset$

.3 $\mathrm{domain}(F) \cup \mathrm{range}(F) = S \cup E$

A3 Auxiliary definitions:

.1 The set $X \triangleq S \cup E$ is called the set of net elements

.2 Given a net element x, we define $x^{\bullet}$ - the <u>post-set</u> of x - and ${}^{\bullet}x$ - the pre-set of x - thus:

$$x^{\bullet} \triangleq \{ y : xFy \} \text{ and } {}^{\bullet}x \triangleq \{ y : yFx \}$$

This concept naturally extends to sets of net elements thus: if A is a set of net-elements then

$$A^{\bullet} \triangleq \bigcup_{A} x^{\bullet} \text{ and } {}^{\bullet}A = \bigcup_{A} {}^{\bullet}x$$

A4 Principal graphic representation

.1 Elements of S are graphically represented by small circles:

○

.2 Elements of E are graphically represented by small boxes:

□

.3 Elements of F are represented as arrows:

.1 ○ → □ , if the element belongs to $S \times E$

.2 □ → ○ , if the element belongs to $E \times S$

A5 Undirected nets

Graphically, an undirected net is represented in the same way as directed nets, except that the arrows which connect circles to boxes and boxes to circles are replaced by line segments with no arrow-heads. Abstractly this may be thought of as a net as defined in A2 with an additional axiom - namely that F is symmetric. The picture $\underset{a}{\bigcirc}-\underset{x}{\square}$ then represents two pairs of the F relation - namely $\langle a, x \rangle$ and $\langle x, a \rangle$.[1]

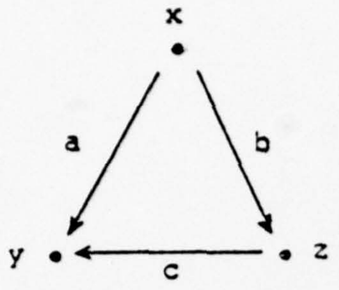
B The General Style of Net Interpretation

The general style in which nets are interpreted is best appreciated in relation to the general style in which graphs are interpreted. Interpreted graphs are most often thought of as <u>relation graphs</u> - i.e. the graph vertices represent a set of elementary objects in some interpreted domain, and the graph arcs represent the elements of a relation among these objects (or a set of relations, if the graph has colored arcs). In "relation thinking" one first thinks of the objects which constitute the field of the relation, accepts them as given, and then thinks of the relation as a construct "over" that field. The formal apparatus reflects this, in constructing the elements of a relation <u>out of</u> the objects (namely, ordered pairs of objects). These relation elements themselves are an abstraction, not thought of in time and space, as the objects themselves usually are.

Now let us approach the "net view" of the world by first seeing how to deform the formal aspect of graphs to become the formal aspect of nets.

We can think of a directed graph as consisting of two primitive entity types: vertices, V , and arcs, A , and a relation $T \subseteq V \times A \cup A \times V$ which we will call <u>touching</u>. The touching relation will not necessarily be assumed symmetric - as it is not in ordinary usage when it is thought of as an act. In the graphic representation we see the touching relation thus:

[1] In a directed net for which the F relation is not anti-symmetric we may find $\underset{a}{\bigcirc}\rightleftarrows\underset{x}{\square}$ which contrasts with the picture $\underset{a}{\bigcirc}-\underset{x}{\square}$ in that it does not assert the <u>necessity</u> of the co-presence of both pairs.



x touches b and a
b touches z
z touches c
a and c touch y

Now we notice that the touching relation in graphs does not condition arcs and vertices symmetrically, as the following axioms show

B1 .1 Every arc touches exactly one vertex (no corresponding bounds for vertices).

.2 Every arc is touched by exactly one vertex (no corresponding bounds for vertices).

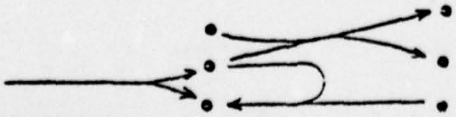
It is clear that the triple $\langle V, A, T \rangle$ satisfies most of the net conditions, as listed in A2. What fails is condition A2.3 because the graph definition permits isolated vertices. Thus a non-empty graph $\langle V, A, T \rangle$ without isolated vertices is a Petri-net with special restrictions - as is also the triple $\langle A, V, T \rangle$.

The net axioms expressed in terms of vertices and arcs would, in place of .1 and .2 merely assert

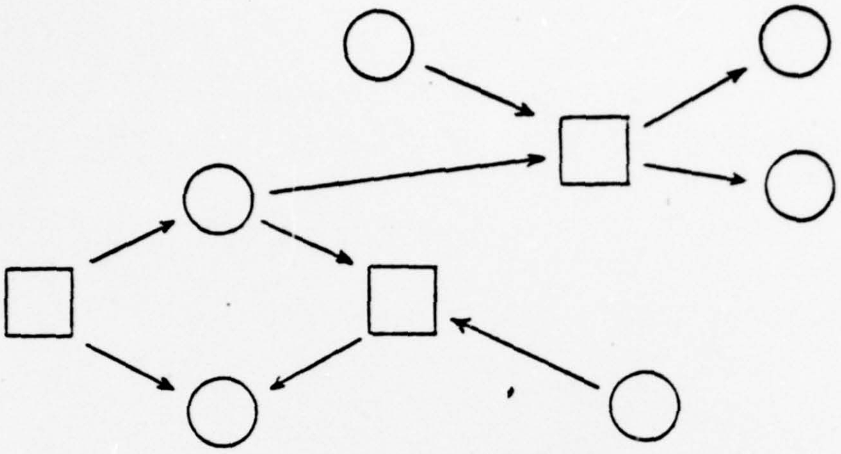
B2 .1 Every arc touches or is touched by at least one vertex

.2 Every vertex touches or is touched by at least one arc

Arcs subject to only restriction B2.1 would differ from the usual arcs in a graph by being potentially "multi-headed" and "multi-tailed" (just as with hypergraphs) as well as possibly lacking a vertex to touch, or to be touched by.



In net graphics the corresponding picture for $\langle V, A, T \rangle$ would be:



Note also that, if the touch relation is forced to be symmetric we then have a natural correspondence between undirected graphs and undirected nets, given by A5 above.

In going from graphs to nets as above, one has to enrich the capabilities of arcs. Since arcs in a graph are normally interpreted as relation elements, what semantic changes in that concept is associated with this enrichment?

B3 .1 Such enriched relation elements are thought of as real, no less, in time and space, than the objects themselves.
Furthermore, as the word "touching" helps to suggest, the relation elements exist in the immediate vicinity of the objects which they hold together - just like a physical coupler which joins several objects together. (Examples follow)

.2 While ordinary relation elements only join two objects to one another - one in the domain to one in the range - the enriched relation elements join n objects to one another. In the context of directed nets the relation-elements thought of as physical couplers have, in general,

k "female" and k' "male" connection points.[1]

Let us now consider examples of net interpretations consistent with the generalities above.

Every example interpretation begins by naming a pair of entity types, one of which is to be thought of as a domain of elements and the other as a domain - one is tempted to say co-domain - of element-couplers - as described above. Since the net axioms A2 are symmetric in S and E one might expect for some, if not all, interpretations that a "dual" interpretation consisting in the reversal of role of the elements and couplers, would make equally good semantic sense.

B4 Examples of Net Interpretations

(Items with asterisks are discussed in some detail below; all items in the list have been developed and used to one degree or another. Some items in the list require discussion to make the connection with nets evident.)

	Element	Coupler
.1	cables	cable-couplers
.2	offices	channels (of communication)
.3	processors	channels (of communication)
.4	* products	(productive) processes
.5	* (organizational) roles	activities
.6	conditions, or states	events[2]
.7	* objects	locations
.8	objects	keys (or descriptors) to objects
.9	structures	constraints[3]
.10	countries	boundaries
.11	propositions	implications

[1] The $k + k'$ objects thus held together by a coupler enter into $k \cdot k'$ pairwise relations with one another in a natural manner. In some developments of Petri-net semantics these pairwise relations are also assumed to be semantically significant.

[2] The state/event interpretation is the source of the letters S and E in the designation $\langle S, E, F \rangle$ for nets.

[3] This interpretation is not immediately apparent, but was put to good use by Genrich [2].

B5 Several comments about the items in this list are in order.

.1 Each item in the list actually represents a class of concrete interpretations, and some items in the list represent very large classes with many notable subvarieties, while others do not. For instance B4.4 (products/processes) could refer to information transforming processes, chemical transforming processes, manufacturing processes, or the productive processes that are referred to in a PERT diagram.

.2 Some of the items in this list are of special importance with regard to the topic of this paper (net-models and organizational systems) while other ones are not. In our descriptions below, we will especially concentrate on the ones that have this importance.

.3 Every item on the list makes the idea of many levels of description of the same underlying reality meaningful. Countries might be grouped together into larger geographic regions whose boundary relations to one another have a systematic relation to the boundary relations among the countries; individual events at some level of description may be grouped together by various principles to form "higher level events" which couple higher level conditions to one another; etc.

.4 In each example <u>some</u> sense of spatial and temporal proximity of elements and couplers can be found, but the sense differs greatly from example to example. With conditions and events, for instance, proximity in space and time refers to the space and time in which the described changes are imagined to take place. With the proposition/implication interpretation (B4.10) the net-topological relations exhibit the proximities of the <u>products and processes of logical reasoning</u> in some particular domain of logical givens - and not necessarily proximities in the world to which the proposition elements refer.

.5 Example interpretations will ultimately be judged by their utilities. The following broad types of utility come into question:

.1 <u>Communication</u> - from a system professional to himself or other;

.2 <u>Formal manipulation</u> - transformations while preserving desired properties as invariants; determining system properties, etc.

.3 <u>Completion</u> - a descriptive discipline can help to call ones attention to missing elements in ones conception of a mechanism, or more generally, system. (E.g. circuit diagrams, where Kirkhoff's Laws apply, have this property.)

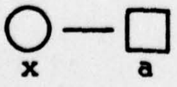
.4 <u>Invention</u> - some representations more predipose to insight and invention than others.

Significant differences in achievements-to-date exist for various classes of net-interpretation. We shall return to this topic at the end.

The first examples we shall discuss are not chosen for their relevance to organizational systems, but rather for the light they shed on the contrast between conventional models and net models.

C <u>'Objects/locations' and 'Regions/Boundaries' Interpretations of Nets</u>

Assume that, at each location some set of objects can be found. Assume further that a single object may be found at several locations because of location overlap. We can now represent a particular distribution of objects over a set of locations by an undirected net, letting elements of S be objects and elements of E be locations. The undirected net connection

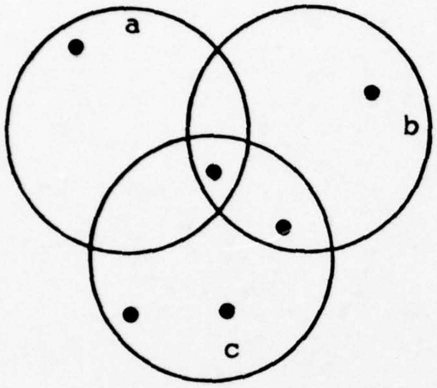


means object x is found at location a .

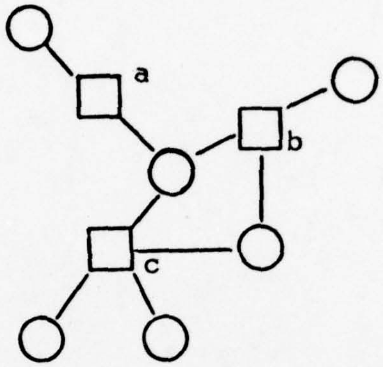
We can also represent such relations in the style of a Venn diagram, drawing dots for objects and areas with smooth boundaries to represent locations. Here is an example, drawn in both styles

C1 Venn diagram (undirected) Net

.1



.2

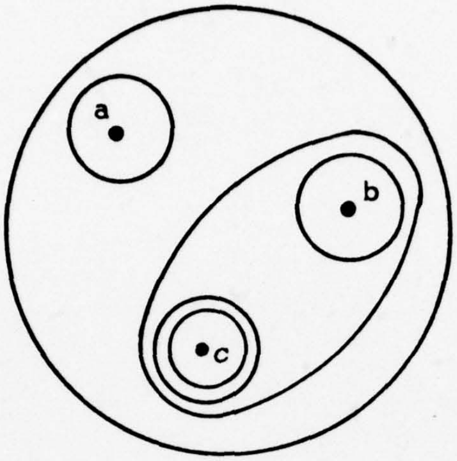


Now Venn diagrams are ordinarily employed to represent point-sets and their boolean relations to one another. By the general style of interpretation as explained above (see especially B3), a net which corresponds to a Venn diagram more naturally represents that Venn diagram than the sets and Boolean relations which the Venn diagram is taken to represent. The diagram is actually made of two kinds of elements - each of them physically substantial ("places" on the paper circumscribed by smooth enclosing lines and dots); the dots in a place all "touch" that place and are "held together" by it in a natural sense.

These intentions in net representations and the manner of their expression have the following interesting consequence in the case of our present example. Without great rigour of mathematical expression, we point to the following well known relations among families of sets, such as are represented by Venn diagrams. Given a set P of points and a family $S \subseteq 2^P$ of sets of points, there is a natural "dual" family of sets, which treats S as a set of points and P as a family of sets of S by the following rule: the point p defines the set $\{ s \in S: p \in s \}$. Given the Venn diagram representation of a family of sets, such as .1 above, we can obtain the diagram for the "dual" family by making a place for each point and a point for each place, and maintaining the "touching" relations unchanged. In the case of C1.1, this transformation yields

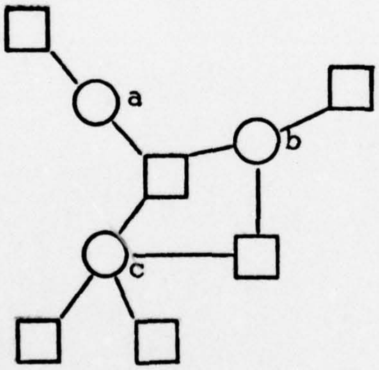
C2

.1



The corresponding transformation of the net C1.2 means interchanging boxes and circles, and nothing else!

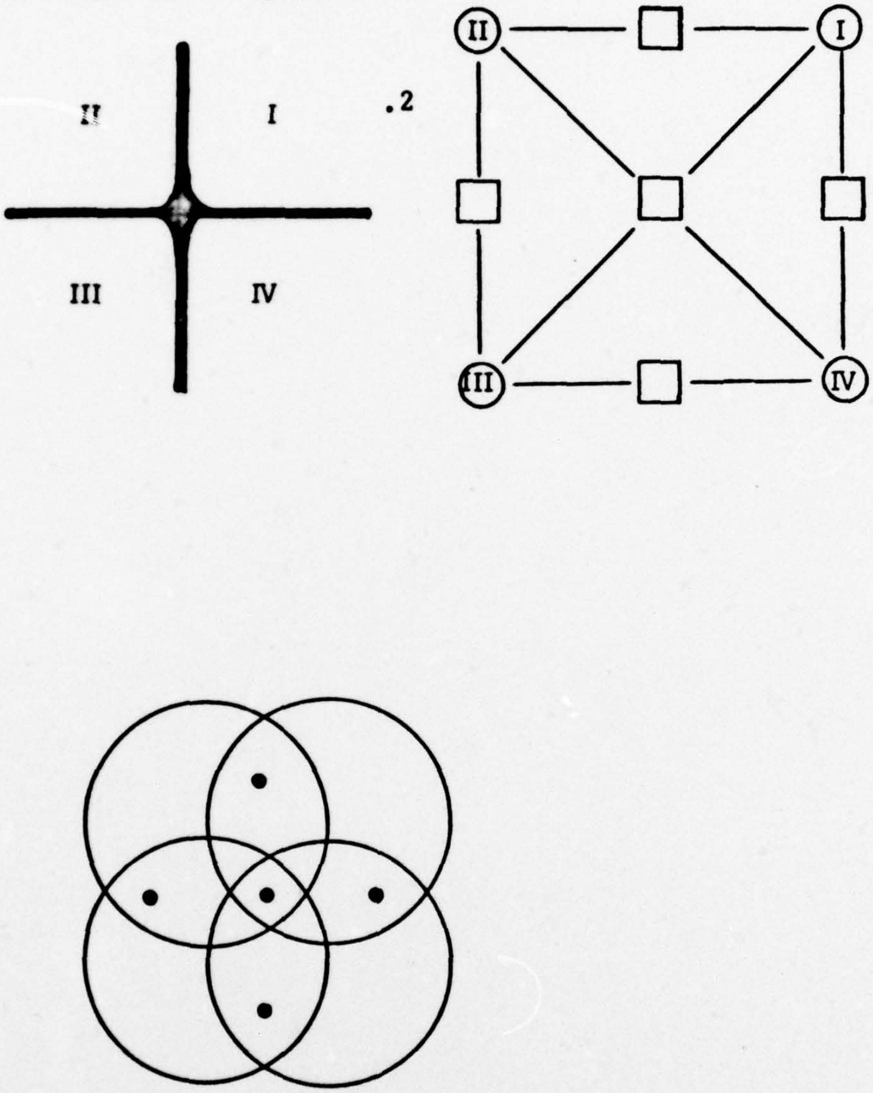
.2



in other words to think of the places as coupled together by the objects instead of thinking of the objects as coupled together by the places. With reference to diagram C1.1 it means seeing the dots as though they were pins which hold sets of discs together, instead of seeing each disc as holding the dots on its surface together. In this way we can understand, in terms of the diagram C1.1, why the nets C1.2 and C2.2 look topologically identical - changed only by the convention of which vertices are squares and which ones circles.

We have now arrived, in a natural manner, at the other interpretation of nets, namely regions/boundaries - or more exactly, regions and boundary elements. Boundaries between regions may be conveniently thought of as couplers between them, much in the same way as the dots in diagram C1.1 may be thought to couple discs together.

C3



.1

.2

.3

The diagrams C3.1 - .3 may be interpreted as symbols for the same idea - the plane divided into four quadrants, as with Cartesian coordinates. The difference between C3.1 and the standard representation of such coordinates lies in the emphasis on the boundaries as substantial. It also illuminates at once the sense in which these boundaries may be parsed into five component elements: the four "spokes" where pairs of regions meet, and the middle, where all four regions come together. Thus the boundary of each region consists of three boundary elements. These relations are re-expressed in net form in C3.2, and finally in the style of the Venn diagrams C3.3 with the "discs" being viewed as coupled to each other by "pins".

Directed nets also come naturally into play in connection with the object/location interpretation by the following route. Locations are useful in locating objects, but objects can be useful in locating locations (as anyone who has used surveyor's stakes must know). Thus, in a given applied context we may wish, for each object x , to distinguish two sets of locations: the pre-set of x , ${}^{\bullet}x$ (defined in A3.2) that is useful in locating x , and the post-set of x , $x^{\bullet}$, that x is useful in locating. Thus we have also made contact with interpretation B4.8 - but further discussion of it lies outside the bounds of this paper.

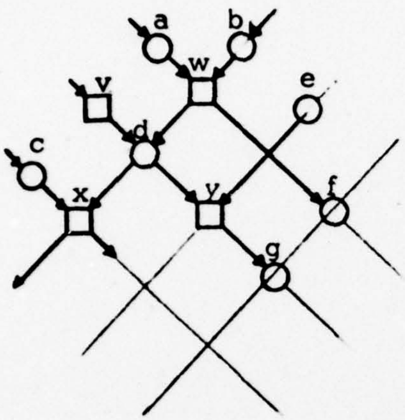
D The 'Products/Processes' Interpretation of Nets

The products/processes interpretation of nets is a large subject which we do not intend to cover in this paper. Here, the purpose is to illustrate net thinking (as in C) in a domain closely related to "organizational systems".

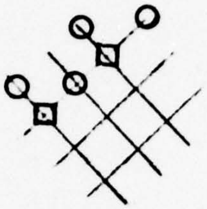
Let us take the E-elements to represent production processes and the S-elements to represent products. For the production process x , ${}^{\bullet}x$ represents the set of products x requires as input, and $x^{\bullet}$ the set of products it generates as output; it follows that, for product a , ${}^{\bullet}a$ represents the set of processes that produce it and $a^{\bullet}$ the set of processes that consume it (or at least employ it).

D1

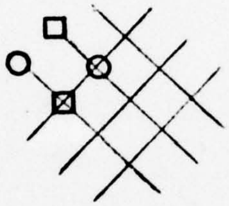
.1



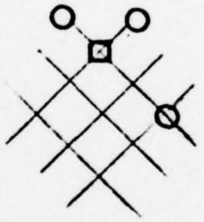
.2



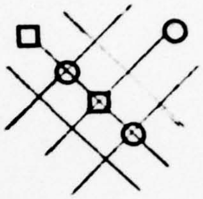
.3



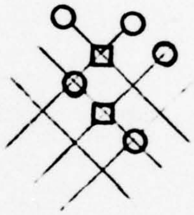
.4



.5



.6



The net fragment D1.1 might be a chart of "what can be made from what by means of what process", such as a chemist or a manufacturer might hang on his wall. If it were chemistry, 'a' through 'g' would be the names of compounds and 'v' through 'y' the names of reactions. From this chart one can at once derive the set of basic recipies for how to reach some end product or end process from the initial ingredients or processes, as shown in the chart. Figures D1.2 - .6 are all the basic recipies to be derived from D1.1; D1.2 and D1.3 are two distinct ones for how to reach process x ; D1.5 and D1.6 are two distinct ones for how to reach product g ; finally D1.4 is for product f .

The formal "dual" of D1.1, obtained by interchanging boxes and circles is, of course, interpretable as another chart of the same type. We have nothing special to say about the meaning of the relationship between two charts that are the formal duals of one another, but consideration of this transformation does help us to understand the meanings expressed in a single chart such as D1.1. We note that the recipies derivable from the dual of D1.1 are not the duals of the recipies D1.2 - .6. (In fact the dual of D1.1 yields eight recipies, not related to any of the recipies D1.2 - .6 by dualization.) The interpreted reason for this is the following: While a process requires <u>all</u> of the products in its pre-set and produces <u>all</u> of the products in its post-set, a product requires <u>any</u> of the processes in its pre-set (in order that it be produced) and helps to enable <u>any</u> of the processes in its post-set. This difference in meaning between S-elements and E-elements is, or course, not expressed in the net axioms A2 alone, but does make its appearance other parts of "net theory".

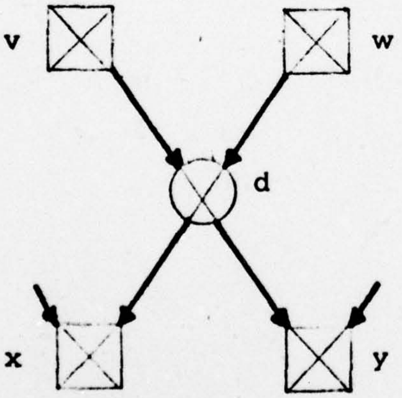
The sample chart D1.1 is free of circuits, although the general interpretation is perfectly consistent with the presence of circuits. (In the case of chemistry, circuits might arise either because of the inclusion of oppostiely oriented reactions - e.g. oxidation as well as reduction - or of processes which maintained one another in dynamic equilibrium.) It is also apparent that everything that has been said so far applied pari pasu to computational processes and computational products.[1]

[1] One of the interesting differences between physical operations (including computation) and operations in the mathematical sense is that the former often produce several results, while the latter, by definition have one result only. That is the reason why, notationally, we so easily slide back and forth between 'a·b' to mean the process of multiplying and 'a·b' to mean the result of that process. This difference affects how big mathematical expressions are formed out of smaller ones versus how big physical structures are formed out of smaller ones. The net expression for an operation under the interpretation discussed in this section is an E-element; it produces a multiplicity of results, as represented by its post-set. Thus it is formally more like a physical operation than a mathematical one.

We would now like to illustrate how net descriptions such as D1.1 may be elaborated to produce new nets that reflect implementation choices. Such elaboration is the now famous "top-down" motion that systems analysts are supposed to practice. An important point about net descriptions is that the relationship between levels - whether produced top-down or bottom-up - are particularly clearly representable. These relationships of composition, decomposition, embedding and excising apply both to products and processes (programs and data). An adequate descriptive (precriptive) discipline must allow one also to relate the elaboration (or compression) in one domain to the same in the other. We are touching upon a large subject. Here we must confine ourselves to small illustrations.

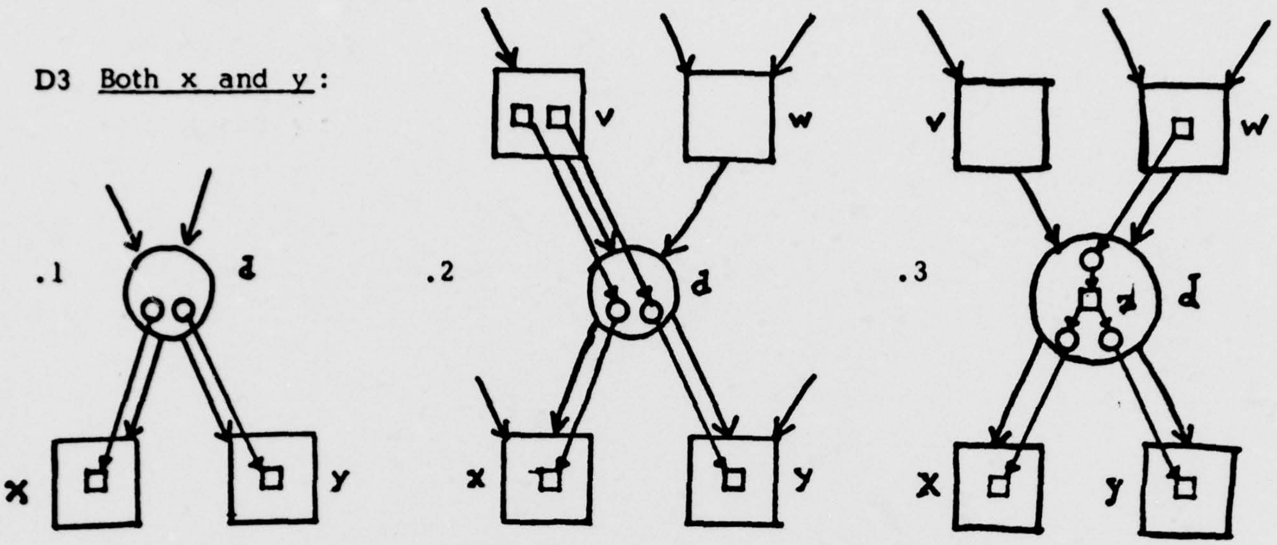
Given a chart such as D1.1 the chemist may wish to describe an experiment or the manufacturer a manufacturing process which will produce, from some subset of initial products, some subset of end produc*s - perhaps with variability in response to anticipated variability of supplies and demands.

D2



The fragment D2 of the chart D1.1 asserts that d is required as input to both x and y . In an implemented process one might want to produce only x , only y , both x and y , or, sometimes x and sometimes y, depending on circumstances.

D3 Both x and y:



The figures D3.1 - .3 exhibit implementation thoughts in the form of net mappings - the dark net with small net elements mapped to the light net with larger net elements. Technically these mappings are continuous - i.e. proximity preserving - with respect to net topology as defined by Petri [ ]. Graphically it is, in each diagram, the small dark net that is mapped to the big light net.

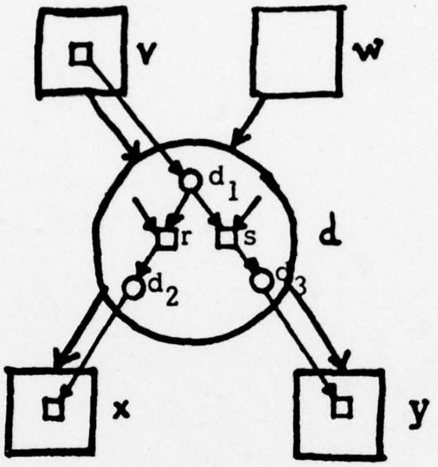
D3.1 divides the product d into two instances of it - instances distinguished by place alone, time alone, or both.

D3.2 proposes the production of both instances of d by two instances of v .

D3.3 proposes the production of d by w , followed by the splitting of d into two new instances which feed x and y respectively. The splitting process may be viewed as a production process whose input is d at one time and place, and whose two outputs are d again, but at different times and places (fan out).

Here is one more example of net elaboration.

D4 Either x or y , depending



In D4 r and s are a pair of especially related processes: r produces the instance of d that x consumes and s produces the instance of d that y consumes. But neither of them produce their output from d alone. They each require, as additional input, a control signal - symbolized in D4 by the extra small input arrows. The intention is that on those occasions that r receives a control signal input s does not, and vice versa. Thus control is exercised over which way d flows on any particular occasion - whether from v towards x or from v towards y .

Some tendencies characteristic of net modeling in contra-distinction to more conventional approaches have been exhibited above, namely:

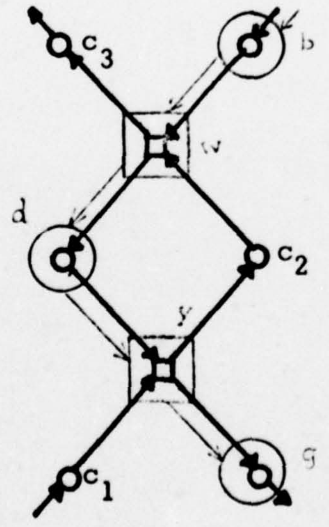
D5 .1 In a 'products/processes' net, two distinct S-elements represent two distinct products and two distinct E-elements represent two distinct processes. In such a net which is the result of elaboration as in D3 - D4 two products may be called different because of where and/or when they are produced, rather than because they are qualitatively different. Thus, in some model of a library facilities two S-elements may be semantically in contrast with one another because they represent two different books, or because they represent two copies of the same book, or because they represent a single book copy at two distinctly defined times and/or places.

In such models the processes of transporting, distributing, retailing and collecting of products (whether in the context of commerce, manufacture or computation) are to be understood as production processes, no less than the processes which transform raw materials into products, in the first place

.2 Derivative of .1 is the tendency to treat control flow as not different in kind from the flow of the products whose movement and transformation is to be controlled.

The explicit introduction of control signals or control quantities into a net of products and processes such as D1.1 is another major motive for the elaboration of such a net. In many contexts the introduction of "control flow" will introduce circuits where none existed before, as the next example shows

D6



The new "products" c_1 , c_2 and c_3 introduced in D6 may represent control signals or control quantities; c_2 in particular governs the rate at which w is to occur, depending on the rate at which y occurs.

We close this section with the mention of several further motives for the elaboration of product/process nets in response to implementation choices.

D7 .1 Equipment scheduling - another class of auxiliary inputs and outputs of processes may be introduced to represent equipment that must be committed to the process and de-committed - or released upon its conclusion

.2 Waste disposal - auxiliary products and processes may be introduced to represent waste products and waste disposal. At certain levels of net description the necessity for the introduction of these products and processes will be forced on the practitioner's attention by formal criteria of completeness of description. (See B5.5.3)

.3 Process (or product) prevention - auxiliary processes and products may be introduced to the model to represent threats of what might occur, or might be yielded if not actively inhibited. A lot of implementation effort has explicitly to do with prevention - such as security measures. But even when prevention is not explicitly at issue, it may implicitly play a crucial role in the definition of systematic relations among processes and products. In D4, for instance, there is the implicit aim of preventing the occurrence of x when y is desired, and preventing the occurrence of y when x is desired. More generally, prevention is an aspect of the exercise of choice.[1]

E The Role/Activity Interpretation of Nets and Organizational Systems

Up to this point we have dealt with domains of interpretation which are part of the stock-in-trade of every systems man - e.g., products/processes. The role/activity concept pair does not have such established standing in systems thinking. It is the direct result of applying nets to the study of "organizational systems", as briefly defined at the beginning of this paper.

Our aim in this section is not only to bring the dimensions of relevance in the role/activity concept pair to view, but also to demonstrate relevance to systems practice. We shall do this with the help of a small example of the following form. A mechanism necessary to the operation of an organization will be specified. With the help of role/activity thinking and net representation of that thinking we shall identify factors that must be taken into account in the implementation of the mechanism. Incidentally this process of identifying factors of implementation may be aptly called "qualitative (system) analysis", in analogy qualitative chemical analysis.

An important aspect of organizational structure is captured by the idea of organizational role. The more formal the organization, the more prominent and elaborated are the roles in terms of which the rules are defined. Military organizations typically have a highly developed formal aspect, and with it, an extensive vocabulary for the classification of roles - e.g. the designations of rank and occupation. Social organization as defined by the law is likewise described in terms of a multitude of roles - e.g. plaintiff, defendant, party to a contract, vendor, customer, witness, judge, etc., etc.

Persons who play roles, we shall call actors. A single person in a single organizational context may play several roles - in sequence, simultaneously, in alternation, etc. In an organizational context some of the defined roles may be played by artefacts instead of persons - any artefact whose state is regarded as significant with respect to the activities which take place. Common items in this category are memory units, and processors, broadly conceived. A door which is locked or not, open or shut, may in certain contexts be formalized as an actor playing one or several roles.

The formal organizational rules specify what it means to play a role. These role specifications include:

Role specifications include:

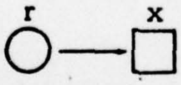
E1 .1 activities required }
.2 activities prohibited } according to rule-defined circumstances
.3 activities permitted }
.4 goals to be aimed at
.5 standards to be maintained

In this paper we shall concentrate on the activity specifications for role players (though an approach for the formal expression of goals has been thought of).

An activity in which an actor engages will result in changes-of-state - his own state vis-a-vis his role, as well as the state of other actors and artefacts which take part in the activity as well.

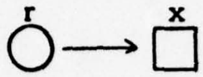
For example the activity of haircutting, engages both a barber and a customer. As a result the customer's appearance improves and he now has the obligation to pay; as another result the barber's floor needs sweeping and he now has the right to be paid.

The haircut example suggests that activites can be thought of as "coupling" roles to one another - haircutting "couples" the barber to the customer. With roles as elements and activities as couplers we can apply the general scheme of net interpretation as described above. In this domain, directed as well as undirected nets are of use. The undirected net connection is taken to mean:

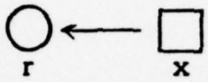


role r participates in activity x

Directed nets arise when we distinguish between roles that participate as givers (of something) and roles that participate as takers of something, relative to a given activity. In haircutting, it is the barber that gives and the customer that takes (the haircut); in the subsequent activity of paying it is the customer that gives and the barber that takes (the money). Thus, the directed net connections have the following interpretations.



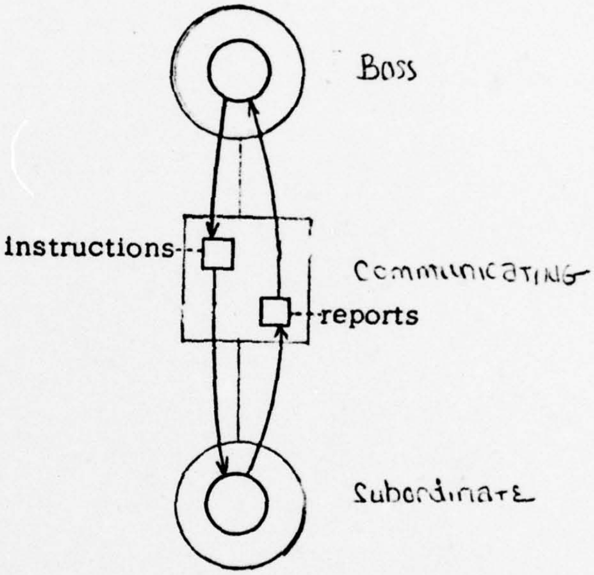
role r is the giver (of something) in activity x



role r is the taker (of something) in activity x

E2 Example

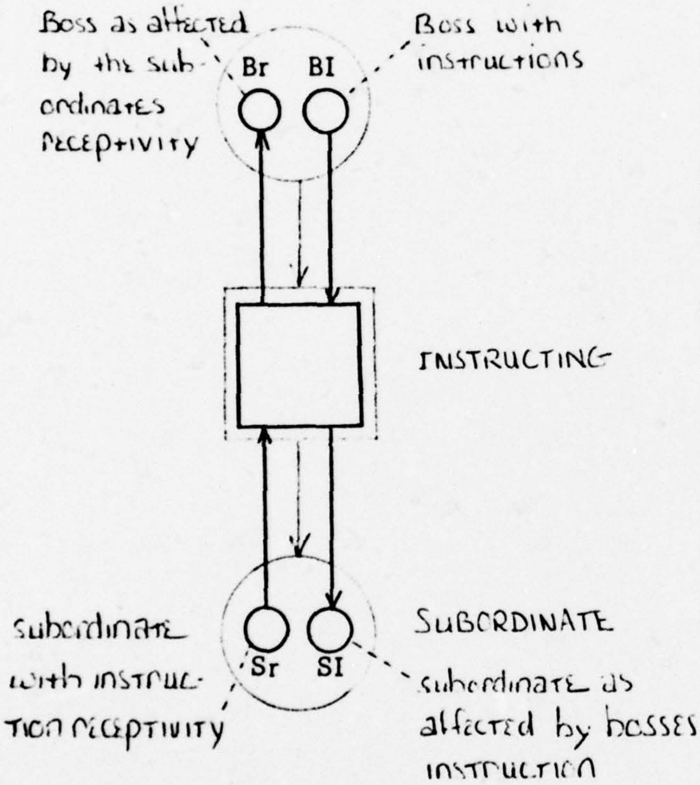
- An undirected role/activity net (drawn large and light)
- A directed role/activity net (drawn smaller and dark)
- A mapping from the latter to the former



It was mentioned above that the result of activity is state change. Associated with each role is a set of states in which an actor who plays that role may be, in a given context, depending on his history of participation in activities. We may think of the states of a role as subroles of it, as will now be explained. Of organizations, we will say that they can have <u>active</u> or <u>inactive</u> states. A company may exist on paper but have no buildings or employees. The active/inactive distinction also applied to the roles that are defined within the organization. (At a greater level of refinement, these may be seen as complex organizations, in turn.) <u>A role is active if there is an actor to play it</u>, <u>and inactive otherwise</u>.

Now suppose that an actor in some role, changes from state a to state b , as a result of participation in an activity x . We may conceptualize this as leaving subrole a - the subrole that permitted him to become engaged in activity x - and entering subrole b , which follows upon completing participation in activity x ; subrole a becomes de-activated, subrole b , activated. With this in mind, we may picture an instruction transfer with the following net construct:

E3


Boss as affected by the subordinates receptivity
Br
BI
Boss with instructions
INSTRUCTING
subordinate with instruction receptivity
Sr
SI
SUBORDINATE
subordinate as affected by bosses instruction

BI "gives" an actor entitled to play the Boss role to Br

Sr "gives" an actor entitled to play the subordinate role to SI .

(partial net mapping)

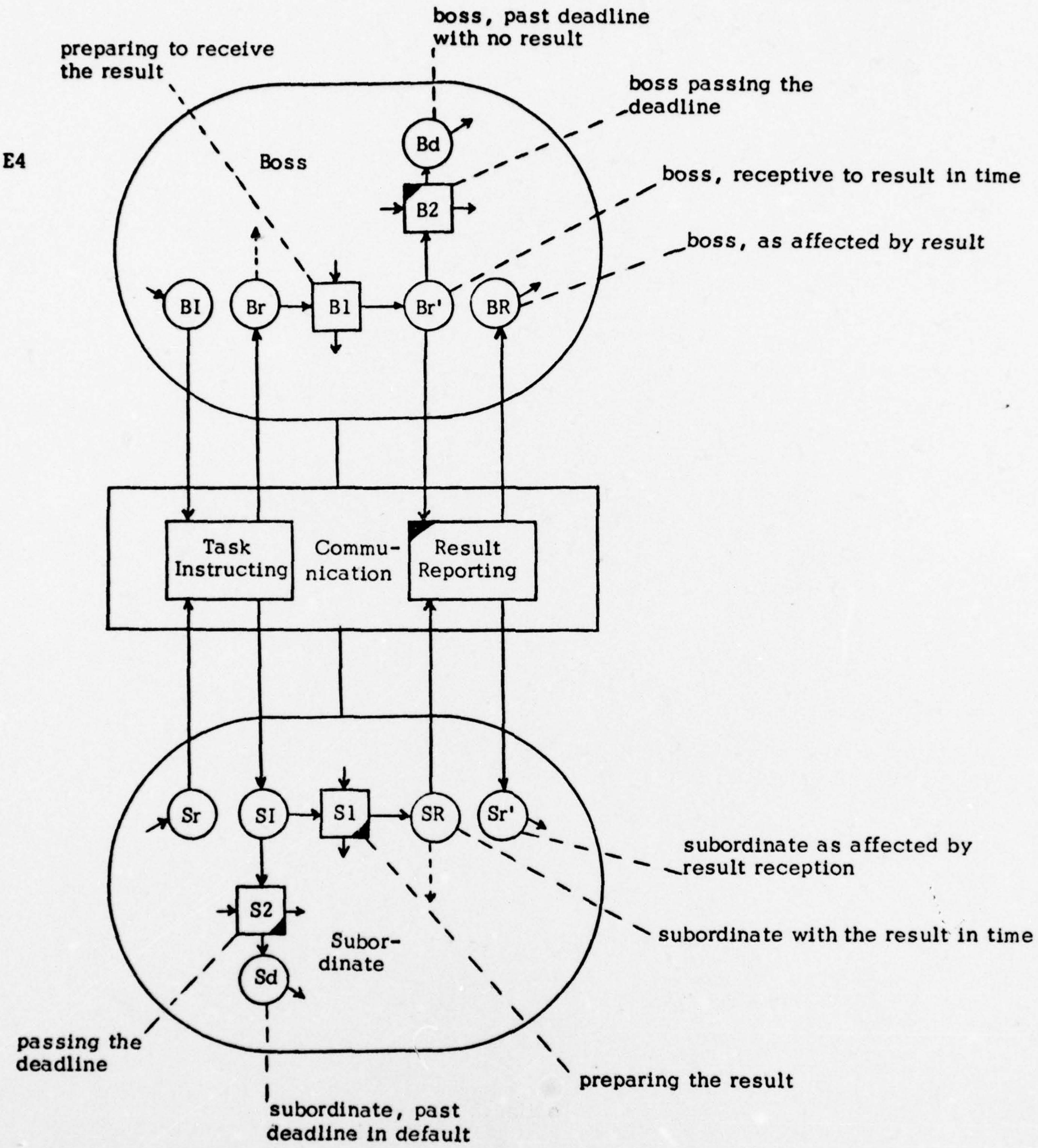
The dark net in E3 can be viewed as an augmented version of the light net to which it maps - augmented by the addition of "back-flow" (Sr to Br), relative to the "forward-flow" of the instructions (from BI to SI). This augmentation is a formal necessity in the building net models at certain levels of description, and leads the systems designer or analyst to notice implementation necessities which he might otherwise forget. For instance, in the absence of the subordinates receptivity cannot occur, no matter how powerful the boss. It may also be important to notice that the boss's formally defined state which follows the state after instruction transfer <u>must</u> be affected by the subordinates receptivity (or lack of it) on that occassion. If not, then the question of how and whether the transfer takes place is of no concern to the boss's role.

Note that the dark net admits the products/processes interpretation: the instructing process uses a boss in state BI and a subordinate in state Sr and produces from them a boss in state Br and a subordinate in state SI .

We are now prepared to construct a somewhat larger (and final) role/activity model of boss/subordinate relations to see what help it offers in understanding implementation requirements ("qualitative analysis").

Suppose the boss sets his subordinate a task and a deadline for its accomplishment. (Note that all real task settings involve - implicitly or explicitly - deadline.) Here is a role/activity model of the critical relations between the boss and his subordinate in this context.

E4


boss, past deadline with no result
preparing to receive the result
boss passing the deadline
Bd
Boss
boss, receptive to result in time
B2
boss, as affected by result
BI
Br
B1
Br'
BR
Task Instructing
Commu- nication
Result Reporting
Sr
SI
S1
SR
Sr'
subordinate as affected by result reception
S2
subordinate with the result in time
Subor- dinate
Sd
passing the deadline
preparing the result
subordinate, past deadline in default

BI, SI, Br and Sr, as in E3.

To understand the content of this model, we must first describe, in a suitable manner, the situation of someone with a deadline which he may or may not meet. Such a person may be thought of as identified with the moving hand of a clock. On the face of the clock a position has been marked as "the deadline". If the person gains access to the resources necessary to accomplish the task in time he ceases to be identified with the clock hand and achieves the state which follows timely task completion. Otherwise he is carried by the clock-drive past the deadline mark into a state of default.

Two such situations are fully pictured in E4. The first is the subordinate in state SI; the second is the boss in state Br'.

Now note that the choice confronting the subordinate in state SI (between *activities S1 and S2 - symbolized as part of a single choice* by the darkened lower corner) must be resolved before the choice confr nting the boss in state Br' can be resolved. The subordinate in position to deliver the result in time (state SR) is the resource the boss needs to become dissociated from his clock hand. On the other hand, the absence of that resource is an essential part of the condition which permits the boss's clock drive to carry him past his deadline. As already remarked earlier in this paper, wherever and whenever a choice between several alternative activities exists, the mere presence of the resources necessary to one of them is not sufficient to enable a positive resolution of the choice; the other alternatives must, at the same time, be inhibited by some absence of resource.

This next series of points, E5 - E8, exhibit a set of implementation necessities which our model helps to reveal.

E5 How much earlier must the subordinate's deadline be than that of the boss? In model E4 that depends on the following factors:

.1 The maximum time allowed for producing the result (S1) - taking variability in the timing of resource availabilities into account. These resources are symbolized in E4 by the extra small arrow entering the box labeled S1.

.2 The maximum expected difference in the rate at which the boss's clock runs and the rate at which the subordinate's clock runs.

.3 The maximum expected reaction time of the boss to the availability of SR and the minimum expected reaction time of the boss to the clock signal which transports the boss into state Bd. (B2).

It is in any case clear that the interval till deadline must be consistent with the just mentioned expectations. Finally, if the boss and subordinate were not directly coupled to one another, but communicated over a transmission line, new factors would enter.

E6 What constrains the time available for the boss to prepare for result receipt?

.1 The minimum expected time for producing the result (S1)

.2 The maximum expected time that the subordinate with the result can remain available - i.e. can remain in state SR.

E7 Failing to produce the result will normally have bad consequences for the subordinate, as mediated by the boss - via state Bd - perhaps by means of the following arrangements. The boss, when once in state Bd is to tap a waiting messenger on the shoulder to dispatch him with bad news for the subordinate - an activity not explicitly represented in E4, but suggested by the small extra arrow pointing out of Bd. For this to work, the messenger must be receptive to a shoulder tap for some period after the time that the boss would enter state Bd, if he were to enter it. Now suppose that in an actual case the subordinate succeeds in delivering the result well ahead of time. <u>The imple-</u>

mentation must assure that during the period after the boss's deadline expiration when the messenger is receptive to a shoulder tap, no shoulder tap is delivered. After all, the mechanism has assured the messenger's receptivity at that time, independent of whether, in actual fact, the subordinate succeeds or not. The real force of these remarks becomes especially visible in a cyclic context where deadlines for the subordinate are repeatedly set and either met or not. But this leads us into aspects of the situation that we can not go into here.

E8 Failure of various durations of state or activity to lie within expected bounds may result in irresolution for the choices we have discussed - in particular, irresolution on the boss's part as to whether he is to pass into state Bd or to pass into state BR (the "Glitch").

We are now ready for closing remarks about the material on net models with the role/activity interpretation.

E9 Responsibility, authority and accountability are all of them aspects of role definition capable of formal expression in the context of nets under the role/activity interpretation, but this has not been demonstrated above. We cannot tell from the very small amount of structure exhibited in E2 - E3 whether the boss is responsible for giving instruction (and if so, to whom), whether he releases them upon demand from his subordinate, or the subordinate accepts them upon demand from the boss, etc., etc. The treatment of these distinctions in role/activity terms cannot be presented within the confines of this presentation.

E10 The relevance of role/activity net models may be questioned, when applied to human organization on the grounds that real bosses and real subordinates relate to each other in a thousand subtle ways which are not governed by explicit or explicatable rules. Even when a business has written rules what is actually done may bear little resemblence to them.

But the point is, we are not here interested in formal role analysis as an aid to the sociologist or psychologist to describe business and governmental behavior. The point rather is to help with the design and construction of mechanized facilities that can be used in the conduct of business and government. In this context the reasons for formalizing certain aspects of the real boss role and certain aspects of the real subordinate role can be at once appreciated if we imagine a boss and a subordinate who have nothing but an electric buzzer to connect them. Now with all other audio-visual channels closed if they are still to cooperate, they will need carefully a worked out code - which means not only syntactic rules that govern symbolic construction with buzzes and silences, but also code of behavior in the accomplishment of their respective tasks - in other words, the formalization of their roles. Th. buzzer is, of course, a metaphore for the forms of communication so highly and complexly mediated which result from the use of computers in organizational settings. Here the need for formalization is particularly pressing because of the computer's ability to "fan in" the effects of many people's actions and "fan out" a resultant as determined by programmed rules to effect many people's actions - a channel with "programmed cross-talk", one might say. In the presence of such communication effects we must introduce rules where none existed before, both in the forms of expression and in the forms of other behavior to which these expressions are connected by meaning.

E11 It was remarked at the beginning of this section that artefacts can play roles, just as people can. This view should not be confused with two others to which it has a superficial resemblance: the view that people and machines are interchangeable in organizational settings, and the more extreme view that people simply are machines.

It was pointed out above that the real roles that people play vis-a-vis one another when they are literally vis-a-vis are not normally confined by rigorous formal definition. In particular such interactions permit scope for the exercise of common sense, which, as we all know, cannot be codified. Artefacts can be depended upon as role players in organizational settings only when the role is fully explicable, as the role of a real person in a responsible position never is.

To comment properly on the view that people simply are machines ("machines made of flesh", as Marvin Minsky once put it) in its relation to our subject and our approach is more than we can do here. At least it should be clear from what has already been said, that our approach does not <u>commit</u> us to this view. A larger discussion woulc reveal that our approach is antithetical to it.

E12 Our business and social life is full of deadlines governed by customs or governed by explicit rule. The timings thus brought into existence are adjusted to each other, and to biological necessities in a complex web of dynamic relations which, for the most part, were not thought out by design, but developed over long periods of time by trial and error. Thus working systems evolved without the benefit of the kind of analytic technique which we illustrated above. There is now, however, a historically new need for such analytic techniques. We now build complex electronic facilities - consisting of interconnected computers, communication links, sensors and effectors - which have become part of the essential fabric of technological society. On the proper function of these systems some of the most vital aspects of well-being and survival have come to depend. In this context the need for <u>explicit</u> understanding of implementation requirements in relation to formal role definitions has become an urgent necessity. When a new town writes its traffic code, there is usually no need for deep thought in advance about the time interval to specify between the serving of a traffic ticket and a demanded response from the accused. When designing a nuclear power plant which must be put out of commission if a signal is not received which says that some test for safe operation has been successfully completed there is plenty of reason for being in intellectual command of the ingredients which govern the time interval, including the ingredients that relate to the human carriers of responsibility in that environment.

E13 Another approach to the use of nets for the modeling of organizational systems has been developed by Petri et/al via the concept pair office/channel. Methods for translating descriptions in the office/channel form into descriptions of the role/activity form have been thought of. A discussion of the circumstances under which one or the other style of description is the more appropriate goes beyond the bounds of this paper.

F What has been Achieved with Net Theory

The use of nets for the doing of practical systems work has only begun. By way of illustration let me cite, by type, a set of applications in which I have myself participated, or that have taken place in my immediate vicinity in course of the last two years.

F1 .1 Computer system design
using state/event and role/activity interpretations
(e.g., geographically distributed access to a common pool of files, with protection against equipment failure at any one geographic site).

.2 Discovering and describing organizational relations, preparatory to system design
using the role/activity interpretation
(e.g., in the City of Boston, for a management information system to improve fiscal control).

.3 System debugging
using state/event and role/activity interpretations
(e.g., curing system death due to time-outs without orderly recovery in a computer net-work application).

.4 Data base design using the object/location interpretation

Although the examples of applied work with nets are promising, they have not yet advanced far enough to demonstrate decisive practical superiority to systems analysis and design without nets. That is, however, another sense in which nets already demonstrate a decisive advantage. Petri-nets as carriers of system meaning are not arbitrary in their form, as other graphical methods for the representation of computing and production processes have tended to be. Each one of the many methods now in use is more-or-less different from its competitors in its choice of primitives and axiomatic constraints, according to the special domain of application from which it evolved or according to the world-view of its orginator, or both, in undeterminable proportions. (An exactly analogous situation exists in the field of programming language .) Such graphical methods, useful as they may be in some engineering context do not further the goal of developing a <u>science</u> of systems. Sciences are always constrained in their forms of expression by "nature", and not by transitory convenience alone.

I firmly believe in the existence of a science of systems. Men can attempt to organize their cooperative or competitive organizational relations to one another in many different ways, but some ways can and do work, while other ways cannot and do not. These possibilities and impossibilities - I believe - are governed by laws as certain as the laws which govern the motions of physical bodies (and are indeed related to these latter).

The concepts appropriate to the expression of these laws - let alone the laws themselves - have not yet been annunciated. For that matter, Computer Science, in spite of its ambitious title, has not put forth much effort towards the development of a scientific basis for its own enterprises. Thus there is, to the best of my knowledge, not even the beginning of a technically adequate answer to the question "what is information, in the context of a system ?" (Nature abhores a vacuum, and so computer science has, faut de mieux, let it seem that information is what the computer processes - a very dangerous view, when coupled to the antecedent and more basic thought that information is what guides men's actions.)

Progress in "net theory" persuades me that it will help to fill the just mentioned gap - more generally, that it will help to shed the light of impartial science upon the sense as well as the non-sense in our current systems practices.

Appendix A

I. Proposal Abstract

A. Research Objective

To produce a step-function improvement in the processes connected with specifying information systems. An information system is defined by a set of lawful relations between consumer/producers of information, whether machine or human. Thus the specification methods should be applicable in the presence or absence of computers. The processes connected with specification that are to be improved are: determining needs; expressing these in a form suitable for procurement of systems; verifying internal consistency of specifications; verifying consistency of performance specifications with implementation specification; documentation.

B. Current Status

The proposed research is based mainly on the earlier work of two groups: the Information Systems Theory Project under A.W. Holt at Massachusetts Computer Associates under the previous sponsorship of ARPA-IPTO and the Institute for Systems Research under C.A. Petri of the Gesellschaft fur Mathematik und Datenverarbeitung, St. Augustin, West Germany. This earlier work has resulted in the development of new concepts and mathematical methods for system specification and analysis connected with a class of mathematical structures called Petri nets. These structures are distinguished from other structures used for similar purposes in the following principal respects: they are tailor-made to represent the dynamic relations between many parts; because of the simplicity of their element relations they lead to large structures that are mathematically tractable; they can be used at gross as well as detailed levels of description with mathematically well-defined relations between levels.

C. The Next Research Steps

The next principal step is to prove utility of the concepts and of the mathematical tools that have been developed in connection with Petri-nets by applying them to parts of real problems connected with information system specification. These applications should result in specification procedure descriptions accompanied by realistic, worked-out examples. They may also result in initial specifications for software that supports the storage, retrieval, and transformation of system specifications.

An auxiliary next step is to improve the mathematical analysis capability relative to Petri-nets, especially with regard to proofs of functionality of suitably represented systems.

II. Technical Proposal

A. Background

This proposal relates to a specific development in the theory and practice of specifying and analyzing information systems. This development, primarily associated with the mathematical/technical invention called Petri-nets, has been centered, in Germany, on the work of C.A. Petri et al since ca. 1962 and, in the United States, on the work of A.W. Holt et al since ca. 1967. At the present time there are some 25 universities and industrial laboratories in the West, and some unknown but substantial number in the East-Block, where research connected with Petri-nets has begun.

Key aspects of the approach include the following:

A1

.1 What is to be specified and analyzed is a set of lawful relations of interaction between a multiplicity of parts - or roles - some of which may be played by people and some by machines. These relations form into a single, meaningful whole, a diversity of producer/consumers of information, each with his own functions and interests.

In this framework of thinking, input/output relations of system parts - so dominant in other approaches to the study of systems - are treated as derivative from requirements of interaction.

.2 Representations at all levels of description - from purpose oriented to implementation oriented - are to have a common formal basis. Different levels of description of the same system are to be related to one another by formally defined "mappings" which aid in understanding these relations and verifying their correctness.

.3 Static constraints - such as typified by data-structure specifications - are to integrate formally with the specification of the intended relations of change - i.e., dynamic specifications. The purpose is to develop methods of verifying that the static specifications and the dynamic specifications are consistent with one another.

.4 Relative to every real system there exist explicit or implied interface requirements which specify the relation between the system and its environment (constraining the behavior of both, relative to one another). Once again the objective is to formally integrate interface specifications with static and dynamic specifications as mentioned above.

A2 The following broad purposes are to be served throughout:

.1 To promote the cause of achieving common understandings between different groups who are concerned with one-and-the-same system - e.g., users, designers, implementers, operators, managers, etc.

.2 To promote the transferability of system designs or components.

.3 To promote the applicability of mathematical techniques for the verification of adequacy of designs, implementations, operation or modification procedures, etc.

All of these broad purposes dictate the emphasis on axiomatic foundations, as per A1.2 above.

B. Comparisons

In the general area of system specification and analysis there are various approaches which, in one respect or another, overlap the intentions expressed above - PERT-CPM, Automata Theory, Structured Programming and Top-Down Systems Analysis, Network-Flow Analysis, various styles of General Systems Theory and Cybernetics, to name but a few. There follow some brief comments about comparisons between Petri-net Theory and the first three items in the foregoing list.

B1 PERT-CPM

PERT-CPM is based on simple, widely applicable, and formalizable concepts that have to do with concurrent, interrelated activities. The technique is principally aimed at the intelligent use of resources with respect to reducing to a minimum the elapsed time of a complex technical project. It is not mainly aimed at the study of cyclic structures of events, or structures in which the choice of what events occur, in what sequences, depends upon environmental influence. Also, while helping to make effective use of resources with respect to project time, the method is not designed to represent resources and their use in a systems context. Explicitly relating the structure (i.e., packaging) of system resources to the structure of events; cyclic (i.e., repetitive) behavior; the transmission of the effects of choice: these are all key issues in the Petri-net approach to the modelling of information systems.

B2 Automata Theory

Automata Theory as well as Petri-net Theory deal with lawful change by discrete means - states and events rather than continuous functions of "time". But in Petri-net theory the formal and interpreted characteristics of "local" states differ significantly from those of "global" states. Spatial dispersion of system parts gives rise to the phenomenon of concurrent operation - a phenomenon which, in the context of automata theory appears as indeterminacy of total system behavior. That is an unfortunate appearance because it cannot be interpreted to mean indeterminacy of system behavior in respect to any of its intended functions. The modelling of concurrency and "system space" have been fundamental concerns in the work on Petri-nets.

While automata theory is more concerned with computation, Petri-net theory is more concerned with relations of communication. While the concept of a computation is naturally related to begin/end, input/output, the concept of communication relation is not. Thus, automata theory - like PERT-CPM - has been mainly interested in linear sequences of events while, in the study of Petri-nets, cyclic structures have been at the focus of attention.

B3 Structured Programming and Top-Down Systems Analysis

Structured Programming and Top-Down Systems Analysis, like Petri-net theory in one of its aspects (see A1.2) are concerned with the relations between levels of description, for the sake of clarity, conciseness and error control. Neither of them are parts of an integrated axiomatic approach to the study of information systems with mathematical depth.

C. Major Research Goals for the Next Three Years

C1 To develop some new tools of demonstrated utility for determining, expressing, and manipulating information system specification.

Such tools would ultimately be embodied in "how-to" manuals and, possibly, in storage, retrieval and manipulation software for information system specification.

The specific areas of improvement over current practice to be achieved are:

C1.1 .1 Improved ability of a specification writer to identify those aspects of his own information system needs that can, and that must be specified.

.2 Improved ability to verify the internal consistency of a set of specifications.

.3 Improvement in communication between specification writer and specification user; in particular an improvement in the ability of a contracting agency to express its information system usage requirements in a way which is contractually binding on a vendor of system designs or constructions.

.4 Improved ability to verify that implementation specifications are consistent with usage specifications.

.5 Improved ability to retrieve the portion of a set of specifications that is relevant to a particular subtask, and to transform it with reference to that subtask.

.6 Improved ability to modify or update specifications in a manner consistent with what remains unchanged.

The emphasis in this phase of the research is on achieving demonstration of utility. This means that the work is to be based on those aspects of Petri-net theory which, in concept, are already far advanced. These are:

C1.2 Techniques of Specification

.1 Role/Activity Analysis of System Requirements Holt [8].

.2 Net mappings (continuous with respect to net topology) as a method for relating levels of specification to one another Petri [9], Hack [4].

.3 Static Structure specification with Petri-nets, Genrich [2].

C1.3 Techniques of Mathematical Analysis

.1 Marked Graph Mathematics Commoner [1], Holt [6], Genrich [3].

Less far advanced, but also relevant are:

C1.4

.1 "Fact" specification, in Petri-nets, Petri [9].

.2 Probability equations and logical dependence analysis as related to nets, Shapiro, Holt [not yet documented] and Holt, Commoner [7].

The first step in accomplishing the subject goal is to relate the existing concepts to actual applied contexts. These contexts should be selected with reference to the Navy's interests,and their potential tractability relative to the present state-of-the-art in net theory. The second step is the construction of "how-to" manuals and the definition of relevant software.

C2 The Mathematical Analysis of the Logical Dependence Relation in Nets

In the last few years there has been developed an interpretation of Petrinets Holt [not yet documented] which permits one to identify dependence relations between decisions - i.e., to observe that the present outcome of a decision X would change if the last outcome of decision Y would change. In the description of such dependence relations, there is also the phenomenon of contingent dependence: the present outcome of decision X depends on decision Y , if the last outcome of decision Z was outcome 1, but not otherwise. As a result of circuit structures in nets, analysis may show that the present outcome of decision X depends on the n^{th}-ago outcome of decision X itself, etc.

The present object of research is to define a useful class of nets in which relations between the outcome of decisions can be computed by algorithm. Preparatory work in this direction has already been accomplished. Success on this front would, in our opinion, have major practical consequences for system science. It would provide a powerful basis for: computations which verify system functionality; the determination of systematic check-out procedures; the design of systems with efficient back up capabilities

D Research Tasks for the Next Year

In re C1

D1 .1 Study an information system to be chosen by agreement between ONR and the contractor in order to define the usage requirements for that system.

.2 Relate these requirements to the systems and procedures now in use to meet these requirements.

.3 Analyze this relation for matches and mismatches between requirements and current practice.

.4 Explore alternative implementations which improve the match.

.5 Report the results of .1 - .4.

In re C2

D2 Proceed on the basis of latest results in the subject area Holt, Commoner [7], Holt [5] and report on results.

1 Commoner, F., Holt, A. W., Evens, S., and Pnueli, A., Marked Directed graphs. J. Computer and Systems Sciences, vol. 5, October 1971, pp. 511-523.

2 Genrich, H. J., The Petri-Net Representation of Structured Knowledge, Proc. MIT Conference on Petri-nets and Related Methods, July 1975, (to appear); also, working paper of the ISF, GMD, St. Augustin, West Germany.

3 Genrich, H. J.; Lautenbach, K.:
Synchronisationsgraphen
Acta Informatica 2, 143-161 (1973)

4 Hack, M., Net Topology, Preliminary Report MIT, 1972.

5 Holt, A. W., Choice in Petri-Nets, a working paper, December 1975, Massachusetts Computer Associates, Inc.

6 Holt, A. W., and Commoner, F., Events & Conditions, vol. 2 of the final report, Marked Graph Mathematics, Contract DAHCO4 68 C 0043, Applied Data Research, Inc., New York, 1971.

7 Holt, A. W., and Commoner, F., Events & Conditions, vol. 3 of the final report, State Machines and Information, Contract DAHCO4 68 C 0043, Applied Data Research, Inc., New York, 1971.

8 Holt, A. W., Role/Activity Models, working paper, July 1975, available from Massachusetts Computer Associates, Inc.

9 Petri, C. A., Interpretations of Net Theory, Proc. MIT Conference on Petri-nets and Related Methods, July 1975, (to appear).

Appendix B

MASSACHUSETTS
COMPUTER ASSOCIATES, INC.
28 PRINCESS STREET, WAKEFIELD, MASS. 01880 • 617/245-9540

December 3, 1976

Mr. Marvin Denicoff
Office of Naval Research
Code 437
800 North Quincy Street
Arlington, Virginia 22217

Dear Marvin:

I realize I am late in communicating with you about 3M, but I have not been idle on that front. This will be a brief progress report and a suggestion for an applied project within my ONR-sponsored activity - a suggestion which arises from the reading that I have done so far in the 3M documents which were forwarded to me from Mechanicsburg. If you approve of the suggestion in principle, I will then proceed to making a more detailed plan.

Thus far, I have studied a document called the Ships Managers 3M Course, and most of volume 1 of the Ship's 3M Manual (OPNAVINST 4790.4). I have also read in other documents, including the SMIP manual. The reading I have done so far fills me with the same mixture of excitement and anxiety which I suppose a young doctor fresh out of medical school must feel when confronting the first serious case for which he must take responsibility. I recognize in this material the attempt to describe real system relations, as I have learned to understand that term through my research, but an attempt which - as I see it - is seriously flawed, for lack of concepts appropriate to the description of systems. The trouble becomes visible at the very beginning, where the 3M System objectives are stated. For example, "... to ensure maximum equipment operational readiness." as an expression of a real system objective, I take to be either meaningless or false. (Note that the intermediate objective F which talks about reducing the costs of maintenance seems either unrelated to the prime objective, or possibly in conflict with it.)

Here, some comments on the general character of system objectives, exemplified by the 3M System. There are two major aspects of system objectives to be distinguished:

A1 those which relate to the relations between the system taken-as-a-whole and its environment;

A2 those which relate to the relations between component parts of the system-relations which are regarded as unchangeable.

Thus the maintenance system taken as a whole must

B1 .1 respond to environmental impacts which result in the need to check the operability of equipment and the need to repair it;

.2 do so with resources which its environment makes available to it;

.3 interface appropriately with constituted agencies, in and out of the Navy - e.g. INSURV, calibration organizations, etc.

On the other hand

B2 the maintenance system involves the cooperative function of constituted agencies with separately defined authorities and responsibilities e.g. Fleet Commanders, TYCOMS, etc., etc. Many of these basic subdivisions of organizational function are not regarded as subject to change in defining the maintenance system. Rather, the keeping in force of these organizational groundlines constitute a part of the "boundary conditions" which the 3M System must satisfy.

One may therefore say that it is part of the objective of the 3M System to permit these various agencies as pre-defined, to accomplish their separate missions. Thus, the objectives of the 3M System cannot be understood without reference to these pre-defined organizational groundlines. (In the Ships Managers 3M Course there appears a list of participating organizations. This list does not make explicit reference to the agencies which are critical in the definition of the interface between the 3M system and its environment. The list also does not reveal what aspect of function of the listed agencies must be supported by the 3M System - and in that sense are relevant to the purpose of the 3M System - and what aspects of function are implementations of the 3M System.)

I have also found, in course of my reading, that at any given level of description there are large gaps and/or irrelevant details - attributable, I believe, to the absence of a disciplined view as to what constitutes a level of description, and how levels of description relate to one another.

These findings, together with an appreciation of the 3M problems that have been identified by the 3M Coordinating Group and Policy Committee (as well as additional problems which I suspect) leads me to suggest that I undertake a new description of the 3M System in its entirety, using the concepts and descriptive techniques associated with net theory. The purposes to be served by the envisaged product are:

.1 to provide improved training material for all classes of personnel connected with 3M (at least four out of the twenty problems cataloged in the SMIP manual point to the need for improvement in this area).

.2 to expose the matrix of causal connections within which system malfunctions (such as those identified in the SMIP manual) are located, and thus provide an improved basis for their solution.

.3 to make easier the recognition of the effects of system change - such as changes in shipboard ADP, in equipments or configurations, in inspection policy, in the organization of overhaul and maintenance facilities, in planning needs, in the organization of the Navy, etc.

.4 to provide improved reference material on 3M procedures, forms, subfunctions, schedules, etc. (Ultimately, a computer-implemented reference facility might be developed).

.5 to improve the orientability of the descriptive material to user concern - e.g. concern with flow of documents, or materials vs. concern with functions and responsibilities; management concern, vs. operational concerns, etc.

.6 to refine the understanding of the boundary conditions which various sub-functions of the 3M System must satisfy -- sub-functions such as MSOD data-processing - to guide their rationalization and improvement.

.7 to provide a model for other system descriptions of relevance to the Navy.

On a very small scale, I have already embarked on this project, in that I have made small role/activity models while reading the 3M materials. The first phase of my effort would, in any case be based on what I can get out of existing documents, but it could not be confined to that. I would certainly need to talk directly with representatives of various Navy organizations, since I believe that, with respect to my objectives, a lost of information is simply missing from the manuals now in circulation. I would also like to establish a standing relationship with a suitable officer from the 3M world, with whom I could discuss my work as it progresses, and who could act as liason between me and that world.

I see this proposed project in the first instance as the applied aspect of my sponsored research, but I would hope that, if it progresses satisfactorily, it could attract additional funding from interested 3M sources.

I would like to close with comments on two key policies which are part of the MDCS philosophy

.1 To make uniform the manner of reporting maintenance activity over the widest possible domain, to promote computer processability, and to promote comparability.

.2 To record all relevant data once-and-only once so that it may then be used by all who have a need of it, in whatever form they require.

The two policies are, of course, related to each other. The basic reporting documents are supposed to be decomposed into such constituent elements that all the various subsequent interests can be served by suitable sub-selection, aggregation, etc. Now I would like to draw attention to several matters that bear on the manner and extent to which these policies should be pursued. First, a highly standardized and widely applicable manner of reporting militates against adaptability in the recognition and the taking into account of new dimensions of relevance in maintenance activities which necessarily arise from time to time - for example, as a result of introducing new technologies. It will also tend to produce irritating overhead burdens in the reporting for any particular class of equipments and systems.

The second is this. The maintenance reporter is in fact, acting as agent for a diversity of interests resident in various organs of the Navy. If he is to be an effective agent, for any one of these organs, he must be able, in some form and at some level, to understand these interests and identify with them. He must be able, in other words, to understand to whom and for what he is responsible. The idea of decomposing the report into a single set of elementary characters which will subsequently be processed into many different messages of which the producer of the report is wholly unaware militates against his understanding of his role. I am of course not saying that the policies .1 and .2 are wrong as such, but only that they need to be tempered by the above-mentioned considerations.

I enclose a copy of a paper which I recently delivered in Germany and which I hope you will enjoy. It should also help to strengthen your understanding of the direction of my effort.

With best regards,

Anatol W. Holt (dtg)

Anatol W. Holt

AWH/dtg
Enclosure